

# A MODULAR PARALLEL-PIPELINED VISION SYSTEM FOR REAL-TIME VIDEO PROCESSING

Publication number: JP2003510669 (T)

Publication date: 2003-03-18

Inventor(s):

Applicant(s):

Classification:

- international: G06T1/20; H04N5/14; H04N9/64; (IPC1-7): G06T1/20; H04N5/14; H04N9/64

- European: G06T1/20

Application number: JP20000511301T 19980908

Priority number(s): US19970058270P 19970908; US19970002265 19971231; WO1998US18608 19980908

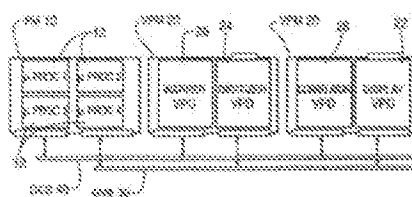
Also published as:

WO9913637 (A2)  
WO9913637 (A3)  
US6188381 (B1)  
EP1013069 (A2)

Abstract not available for JP 2003510669 (T)

Abstract of corresponding document: WO 9913637 (A2)

A real-time modular video processing system (VPS) which can be scaled smoothly from relatively small systems with modest amounts of hardware to very large, very powerful systems with significantly more hardware. The modular video processing system includes a processing module containing at least one general purpose microprocessor which controls hardware and software operation of the video processing system using control data and which also facilitates communications with external devices. One or more video processing modules are also provided, each containing parallel pipelined video hardware which is programmable by the control data to provide different video processing operations on an input stream of video data. Each video processing module also contains one or more connections for accepting one or more daughterboards which each perform a particular image processing task. A global video bus routes video data between the processing module and each video processing module and between respective processing modules, while a global control bus provides the control data to/from the processing module from/to the video processing modules separate from the video data on the global video bus. A hardware control library loaded on the processing modules provides an application programming interface including high level C-callable functions which allow programming of the video hardware as components are added and subtracted from the video processing system for different applications.



Data supplied from the *espacenet* database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2003-510669

(P2003-510669A)

(43) 公表日 平成15年3月18日 (2003.3.18)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テ-マ-ド* (参考)
G 0 6 T 1/20		G 0 6 T 1/20	C 5 B 0 5 7
H 0 4 N 5/14		H 0 4 N 5/14	Z 5 C 0 2 1
9/64		9/64	Z 5 C 0 6 6

審査請求 未請求 予備審査請求 有 (全 75 頁)

(21) 出願番号 特願2000-511301(P2000-511301)  
 (86) (22) 出願日 平成10年9月8日 (1998.9.8)  
 (85) 翻訳文提出日 平成12年3月8日 (2000.3.8)  
 (86) 国際出願番号 P C T / U S 9 8 / 1 8 6 0 8  
 (87) 国際公開番号 W O 9 9 / 0 1 3 6 3 7  
 (87) 国際公開日 平成11年3月18日 (1999.3.18)  
 (31) 優先権主張番号 6 0 / 0 5 8 , 2 7 0  
 (32) 優先日 平成9年9月8日 (1997.9.8)  
 (33) 優先権主張国 米国 (U S)  
 (31) 優先権主張番号 0 9 / 0 0 2 , 2 6 5  
 (32) 優先日 平成9年12月31日 (1997.12.31)  
 (33) 優先権主張国 米国 (U S)

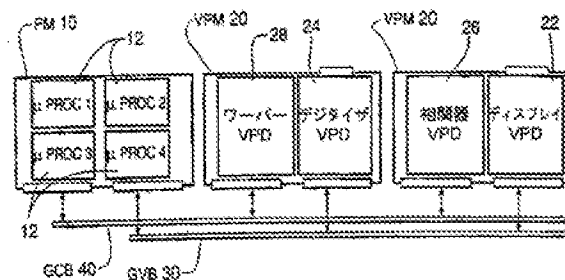
(71) 出願人 サ-ノフ コ-ポレイション  
 アメリカ合衆国、ニュージャージー州  
 08543、プリンストン シーエヌ 5300、  
 ワシントン ロード 201  
 (72) 発明者 ファン デール ヴァル、 グーイツェ  
 ン、 エス。  
 アメリカ合衆国、ニュージャージー  
 州、ホープウェル、ダブリュー、ブ  
 ロスペクト ストリート 105  
 (74) 代理人 弁理士 山田 行一 (外1名)

最終頁に続く

(54) 【発明の名称】 リアルタイムビデオ処理のためのモジュラ平行パイプライン化ビジョンシステム

## (57) 【要約】

適度のハードウェア量を有する比較的小さいシステムから、かなり多くのハードウェアを有するかなり大きく強力なシステムに、滑らかにスケーリングすることができるリアルタイムモジュラビデオ処理システム (V P S)。制御データを使用してビデオ処理システムのハードウェア及びソフトウェア操作を制御し、外部デバイスを有する通信を促進する少なくとも1台の汎用マイクロプロセッサを含む処理モジュールを含むモジュラビデオ処理システム。1つ以上のビデオ処理モジュールが提供され、制御データによってプログラム可能な並列パイプライン化ビデオハードウェアを各々含み、ビデオデータのインプットストリームの上で異なるビデオ処理操作を提供する。各ビデオ処理モジュールも、各々特定のイメージ処理タスクを実行する1つ以上のデータボードを受け入れるために1つ以上の接続を含む。グローバルビデオバスは、処理モジュールと各ビデオ処理モジュールとの間、及び各々の処理モジュール間でビデオデータをルーティングし、グローバル制御バスが、処理モジュールへ/からグローバルビデオバスのビデオデータから分離



**【特許請求の範囲】**

**【請求項1】** モジュラビデオ処理システムであって、

制御データを使用して前記ビデオ処理システムのハードウェア及びソフトウェア操作を制御する少なくとも1台の汎用マイクロプロセッサを含んでいる処理モジュールと、

ビデオデータのインプットストリームの上で異なるビデオ処理操作を提供するために前記制御データによってプログラム可能な並列にパイプライン化されたビデオハードウェアを含む少なくとも1つのビデオ処理モジュールと、

前記処理モジュールと前記少なくとも1つのビデオ処理モジュールとの間でビデオデータをルーティングするグローバルビデオバスと、

前記グローバルビデオバス上の前記ビデオデータから別個の前記少なくとも1つのビデオ処理モジュールから又はそこへ前記処理モジュールに又はそこから前記制御データを提供するグローバル制御バスと、を含むシステム。

**【請求項2】** 前記ビデオデータがシステムクロックに同期する関連ビデオタイミング情報に結合し、各ビデオ処理モジュールが前記ビデオデータ及び関連ビデオタイミング情報を各々の並列パイプライン化ビデオハードウェアコンポーネントに又はそこからルーティングするクロスポイントスイッチを含み、前記タイミングデータが前記ビデオ処理モジュールのパイプラインの遅れを補償している請求項1記載のシステム。

**【請求項3】** 少なくとも1つのビデオ処理モジュールが、前記ビデオデータ及び関連ビデオタイミング情報に応答する可変な演算論理ユニット（CALU）を含み、それによって各々のイメージ間のインプットビデオビデオタイミング中の差を自動的に補償するようになっており、デュアルイメージポイントワイズビデオ処理操作及びイメージ蓄積を提供する請求項2記載のシステム。

**【請求項4】** 少なくとも1つのビデオ処理モジュールが、異なる解像度でビデオデータのピラミッド表現を発生させる少なくとも1つのピラミッドフィルタリングプロセッサを含み、それによって前記ビデオ処理システムの前記特定のイメージ処理タスクのために前記ビデオデータのリアルタイム処理促進する請求項2記載のシステム。

【請求項5】 各ビデオ処理モジュールが少なくとも1つのデータボード接続を含み、該データボードは前記ビデオデータ上にビデオ処理関数を実行し、該処理関数は前記ビデオ処理システムの特定のイメージ処理タスクに特有である請求項2記載のシステム。

【請求項6】 前記処理モジュールが少なくとも2台のマイクロプロセッサを含み、各プロセッサが他のどれかのマイクロプロセッサと共有されない関連ランダムアクセスメモリと、各マイクロプロセッサから前記共用メモリのを任意にリクエストする任意制御バスを通して前記プロセッサモジュールの各マイクロプロセッサでアクセス可能な共用メモリと、を有する請求項1記載のシステム。

【請求項7】 前記処理モジュールの汎用マイクロプロセッサ上でロードされるハードウェア制御ライブラリを更に含み、前記ハードウェア制御ライブラリが、所定の処理操作を実行するための前記少なくとも1つのビデオ処理モジュールの並列パイプライン化ビデオハードウェアをプログラムするための関数群を含む、請求項1記載のシステム。

【請求項8】 グローバル制御バスに制御データを使用して前記ビデオ処理システムのハードウェア及びソフトウェア操作を制御する少なくとも1台の汎用マイクロプロセッサを含む処理モジュールを接続するステップと、

前記グローバル制御バスに、前記制御データによってプログラム可能な並列パイプライン化ビデオハードウェアを含む少なくとも1つのビデオ処理モジュールを接続し、ビデオデータのインプットストリームの上で異なるビデオ処理操作を提供するステップと、

前記処理モジュールが、前記グローバル制御バスに接続された各ビデオ処理モジュールの存在を検出するステップと、

前記処理モジュールが前記制御データを各検出されたビデオ処理モジュールに前記グローバル制御バスを越えてパスし、前記並列パイプライン化ビデオハードウェアプログラムして前記ビデオ処理システムの特定の処理タスクに特有なビデオ処理関数を実行するステップと、を含むモジュラビデオ処理システムをつくる方法。

【請求項9】 グローバルビデオバスに、システムクロックに同期した関連

ビデオタイミング情報を有する前記ビデオデータを結合させる追加のステップと、

クロスポイントスイッチを介して前記グローバルビデオバス越えて前記ビデオ処理モジュールの各々の並列パイプライン化ビデオハードウェアコンポーネントに又はそこから関連ビデオタイミング情報及び前記ビデオデータをルーティングする追加のステップとを含み、

前記タイミングデータが、前記ビデオ処理モジュールでパイプラインの遅れを補償している請求項8記載の方法。

【請求項10】 データのインプットストリーム上に異なる専門化された処理操作を提供するためにプログラム可能な並列パイプライン化ハードウェアを含む少なくとも1つの専門化された処理モジュールと、

前記専門化された処理モジュールのハードウェア及びソフトウェア操作を、前記汎用マイクロプロセッサ上にロードされるハードウェア制御ライブラリを使用して制御する汎用マイクロプロセッサを含む汎用処理モジュールであって、前記ハードウェア制御ライブラリが前記少なくとも1つの専門化された処理モジュールの前記並列パイプライン化ハードウェアをプログラムするための関数群を含み、所定の専門化された処理操作を実行する処理モジュールと、

前記汎用処理モジュール及び前記少なくとも1つの専門化された処理モジュールによって処理される前記インプットデータと別個の前記少なくとも1つの専門化された処理モジュールから又はそこへ前記汎用処理モジュールの前記ハードウェア制御ライブラリに又はそこから制御データを提供するグローバル制御バスと、を含むモジュラ処理システム。

【請求項11】 前記インプットデータがシステムクロックに同期する関連したタイミング情報に結合され、各専門化された処理モジュールが、ルート前記インプットデータ及びその関連したタイミング情報を、各々の並列パイプライン化ハードウェアコンポーネントに又はそこへルーティングするクロスポイントスイッチを含む、請求項10記載のシステム。

【請求項12】 前記ハードウェア制御ライブラリが前記専門化された処理システムの各ハードウェアコンポーネントの装置情報を含み、前記ハードウェア

制御ライブラリの前記関数が前記装置情報を操作して、前記異なる専門化された  
処理操作の各々の前記ハードウェアコンポーネントをプログラムする請求項10  
記載のシステム。

**【発明の詳細な説明】****【0001】****【政府関与の声明】**

本発明は、政府契約番号第D A A K 7 0 - 9 3 - C - 0 0 6 6 号の下なされた。政府は、本発明において一定の権利を有する。

**【0002】****【関連する出願に対するクロスリファレンス】**

本出願は、1997年9月8日に出願の米国仮特許出願連続番号第60/058,270号の利益を主張する。

**【0003】****【発明の背景発明の分野】**

本発明は、リアルタイムビデオ処理の並列パイプライン化ビジョンシステム及びビデオ信号を処理するための対応する方法に関し、より詳細には、モジュラ（モジュール式，modular）に設計され、異なるアプリケーションに容易にスケラブルなシステムに関する。

**【0004】****【従来技術の説明】**

データキューブ(Datacube)によって分散且つ開発されるマックスビデオ250システムは、複数のビデオチャネルの処理ができるシステムの1つの例であり、20M p i x e l / s e cビデオレートで、各々ピクセルを処理することができる。しかし、MV250は単に単一のフィルタリング部材を含み、一方、多くがしばしばリアルタイムマルチ解像度イメージ処理を必要とする。MV250は、マルチ解像度イメージの上で動作する限られた性能を提供する。それは単に5つのタイミングチャネルを有し、ビデオクロスバーへの32インプット及びアウトプットバスがあるからである。加えて、各処理パスのための適切な遅延を計算するためにはかなりの量の制御オーバーヘッドが必要である。更に、ビデオ処理マザーボード間には、ビデオバスの単に限られた量のみが利用可能で、ワンボードの上で得られる5つのタイミングチャネルは、全ての処理ボードで分配される。その結果、マルチ解像度処理の上でスケーリングは、かなり厄介になり、利用可能資

源の使用は、能率が悪い。

#### 【0005】

そのMV250も、システム制御用の標準汎用プロセッサボードの使用法に依存し、従って、処理されたビデオデータを伝達するための手段を、ハイレベルビデオデータ処理用汎用プロセッサに効率的に提供しない。データキューブは、そのようなビデオインタフェース（例えば1960ベース処理ボード）を有する特殊目的DSPボードを提供することによって、速いビデオデータストリーム搬送を有するシステムに汎用処理性能を加えるための特殊な解決を提供する。しかし、その場合には、ビデオハードウェアの制御及びビデオデータのハイレベル処理は分離され、アルゴリズム実施の効率がかなり低減され、タスクを同期させるために追加の制御ソフトウェア及び制御時間が加えられる。

#### 【0006】

イメージングテクノロジー150/40は、並列パイプライン化システムの他の例であり、40Mpixels/secを処理することができ、また、同時に多重チャンネルを通ることができる。このシステムは、ビデオ処理ボード内のビデオデータバスの分散及びビデオ処理ボードの中が、更に限られていること以外MV250と同様の限界を有する。

#### 【0007】

そのような専用のパイプライン化ビデオ処理システムによる効果的なリアルタイムビデオ処理は、「フロントエンド」操作と呼ばれる多数の操作の効果的な実行を必要とする。リアルタイムビデオ処理は、所望の速度でよりハイレベル操作を実行するためにこれらのフロントエンド操作に非常に依存している。主なフロントエンドの操作のうちの1つは、マルチ解像度イメージ表現の発生である。これらの表現は、一般にイメージピラミッドとして言及され、低い空間分解での連続表現へのそのフル解像度でのオリジナルイメージの分解を含む。これは、反復したイメージのフィルタリング及びフィルタリングされた結果のサブサンプリングを通して実行される。最も一般的なピラミッドは、ガウスピラミッドと呼ばれており、連続するローパスフィルタリングを含みオリジナルイメージの10分の1を取ってより低い空間分解でイメージ特徴を表す、より小さいイメージシーケン



スを提供する。そのようなピラミッドフィルタリングを提供するピラミッドプロセッサ集積回路は、例えば、米国特許第5,359,674号及び米国特許出願連続番号第08/838096号の発明者のうちの1人によって説明された。

#### 【0008】

効果的なリアルタイムビデオ処理システムは、リアルタイムビデオレートでフロントエンド操作を実行し、フロントエンド処理の結果を、その結果を分析してその結果に基づく決定を作る汎用プロセッサに提供することができなければならない。しかし、フロントエンドプロセスに続くハイレベル操作は、通常かなり特殊なアプリケーションでかなり複雑である。これはハードウェアの最適化のために好適ではないよりハイレベルの操作をつくる。他方、フロントエンドのプロセスは、遍在しており、ハードウェアで効率的に実行可能でなければならない。

#### 【0009】

これらの考慮事項に基づいて、特徴の以下のリストは、効果的なリアルタイムビデオ処理システムを定義するために使用されることができる：

速いコンボリューション(convolution)及びピラミッド発生。これは、勾配フィルタ及び他の一般に適用できるフィルタリング演算と同様に、ガウス及びラプラシアンピラミッドの発生を含む。

#### 【0010】

再構成可能な演算論理ユニット。イメージポイントワイズ加算、減算、乗算及び他のより任意の操作は、フロントエンドプロセスに対してかなり一般である。

#### 【0011】

ルックアップテーブル操作。これらの操作は、イメージにポイントワイズで行される単一イメージ変換を含む。イメージにゲインを加えること、イメージを逆にすること、イメージをスケールリングすること、イメージをスレッシュホールドすること、及びそのような他の関数はルックアップテーブル変換の典型である。

#### 【0012】

効果的な並列アーキテクチャ。これは、効果的な方法で並列に複数のコンポーネントを使用する能力を説明する。処理リソースは、処理リソースが使用中の間、柔軟に使用されることができない場合は有用でない。

## 【0013】

汎用プロセッサへのビデオデータの速い移動。イメージデータがDSP又は汎用マイクロプロセッサで分析されなければならない場合、イメージデータは、速くアクセスできなければならない。

## 【0014】

ハイレベルハードウェア制御。再インプット可能なマルチタスキングの環境が、最高効率及びプログラム化可能性を達成するためにハードウェア制御に利用できなければならない。

## 【0015】

本願の譲り受け人によって開発され、センサVFE-100として既知のリアルタイムビデオ処理システムは、これらの考慮事項を考慮して開発された。VFE-100は、リアルタイムイメージスタビリゼーション、動きトラッキング、変化検出、ステレオ視、シーン内の関心対象の速い検索、ロボットの誘導、などを、ピラミッドフィルタリング技術を使用して各シーンで重要な要素に集中することによって提供し低減された解像度及びサンプル密度で最初の処理を実行し、必要に応じてより高い解像度で処理を次第にリファインする。

## 【0016】

VFE-100によるビデオイメージ処理は、3つのステージで生じる。即ち、信号変換、信号選択及び計数値推定。2つの基本的信号変換は、サポートされる。初めに、イメージワープは、一般の座標系にイメージのペアを持ってきて（動き、ステレオ又は観察及び参照イメージ）後続のプロセスは、均一及び局所的になるであろう。第二に、ピラミッド変換は、空間ドメインのイメージ信号をバンドパスコンポーネントに分解する。信号選択は、ピラミッド変換ドメインで実行される。後続の解析で使用するデータの解像度及び位置は、ピラミッドの適切なレベルで、適切なウィンドウからそれらのデータを選択することによって制御される。選択されたデータは、関心の計数値の推定値を得るために次に処理される。これらは、動き又はステレオの場合変位ベクトルを含み、又は、変更及び或る方向に向けるためのテクスチャー又は特徴エネルギーを含む。変換されたデータは、例えばローカル相関性、自由度、テクスチャーエネルギー及びローカルモー

メント等の測定を説明することができるコンパクトフィルタのアプリケーションを通して解析に合うようにフォーマットされる。

#### 【0017】

VFE-100は、リアルタイムビジョンアプリケーション用汎用コンピューティングエンジンとして設計される。パイプライン処理は、要素のシーケンスを通したイメージデータフローとして実行され、データフローパス及び処理要素は、多種多様な作業を供給するために変更されることができる。構成されると、ステップのシーケンスは、外部制御なしで全イメージ又はイメージのシーケンスのために実行されることができる。

#### 【0018】

モジュラで、ハードウェアの適度の量を有する比較的小さいシステムから、かなり多くのハードウェアを有する、かなり大きくかなり強力なシステムまで滑らかにスケーリングするように、そのようなリアルタイムビデオ処理システムを設計してVFE-100を改良することは望ましい。特に、新しい処理コンポーネントの加算及び利用できる装置の再構成を通して、特殊なアプリケーションにカスタム調整されうるモジュラのリアルタイムビデオ処理システムを設計することは望ましい。本発明は、当該技術のこれらの要件を満たすように設計された。

#### 【0019】

##### 【本発明の概要】

当該技術での上記の必要性は、本発明の実施形態に従って、VPSが、ハードウェアの適度の量を有する比較的小さいシステムから、かなり多くのハードウェアを有する、かなり大きくかなり強力なシステムまで滑らかにスケーリングされることができるようにするモジュラのビデオ処理システム(VPS)を提供することによって対処される。本発明の実施形態のVPS設計のモジュラ性は、新しい処理コンポーネントの加算及び利用できる装置の再構成を通して、VPSが特殊なアプリケーションにカスタム調整されることができるようにする。

#### 【0020】

特に、本発明は、リアルタイムビデオ処理システム及び処理ビデオ信号のための方法である。本発明の好ましい実施形態に従って、VPSは、マイクロプロセ

ッサ（好ましくはデジタルシグナルプロセッサ（DSP））及びハードウェア制御ライブラリ（HCL）を含む最高2つの追加のデータボードと、特殊アプリケーションボードを有する1つ以上のモジュールのパイプライン化されたビデオ処理モジュール（VPM）と、PM及びVPMを接続しているグローバルビデオバス（GVB）及びグローバル制御バス（GCB）を有するモジュールのマルチプロセッサ処理モジュール（PM）に分解される。

#### 【0021】

本発明の実施形態のVPSは、1つ以上のVPMを含むモジュールの構成で、多数の要素を結合しており、複数のビデオデータストリームの上で効率的に動作する基本的処理関数を提供し、複数のビデオデータストリームを保存することができ、且つ検索することができる。「フロントエンド」ビデオ処理に対して、例えばフレームストア、2Dフィルタ関数、ルックアップテーブル（LUT）操作及び1及び2のインプット算術演算等の基本的ビデオ処理要素の倍数が必要である。

#### 【0022】

ビデオ処理要素及びビデオデータストリームが、並列して困難なく、効率的に変数イメージ解像度のビデオデータを扱うので、各ビデオデータストリームは異なるイメージ解像度だろう。好ましい実施形態で、それは、前記の米国特許第5,359,674号及び米国特許出願連続番号第08/838096号において説明した技術を使用して各ビデオストリームと共に独立の時間情報を提供することによって実行される。制御諸経費をかなりを低減する追加の利益有する。それは、パイプラインの遅れが各処理操作のために計算される必要がなく、多くの関数は、自動的に時間情報に基づく彼らのパラメーターを適合させることができるからである。

#### 【0023】

本発明の実施形態のVPSは、好ましくは、様々な小さく特殊なビデオ処理モジュール、又はビデオ処理データボード（VPD）を含み、それはより多くの特殊アプリケーション関数を実行するようになっている。一般的に、VPDは、それで、その複数のVPDは、機械的に各VPMに取り付けられることができるよ

うに実行される。VPDは、例えばビデオデジタイザ又はビデオディスプレイ等の、かなり特殊なビデオ処理関数から、例えばイメージ相関性及びイメージワーピング又は特殊なイメージストレージ関数の専用ビデオ処理関数の範囲に値域を定めること、又は、1つ以上のDSPのような汎用プロセッサを含むことができる。本発明の実施形態のVPSは、最高4つのプロセッサを有する少なくとも1つの汎用処理モジュール(PM)を含み、VPSで全ての関数を制御し、外界に制御及び/又はデータインタフェースを提供し、VPSのビデオ処理関数によって最初に処理されたデータに汎用処理を提供するタスクを有する。VPSは、効果的なマルチプロセッサマルチタスク及びマルチスレッディング技術の他に、効果的なマルチプロセッサアプリケーションのための汎用プロセッサの数をスケールリングするための方法を好ましくは提供する。

#### 【0024】

グローバルビデオバスは複数のVPMがシステムに加えられるとき、バンド幅でスケールリングすることができる効果的且つ高バンド幅のビデオストリームデータバスで、グローバル制御バスはPMとVPSのレストとの間で効果的な制御バスである。グローバル制御バスは、PMのハードウェア制御ライブラリからVPMまで制御データを通過させVPMの処理関数の多くを実行するために使用されるフィールドプログラマブルゲートアレイの高速のプログラミングをできるようにする。グローバル制御バス上の制御データの、グローバルビデオバス上に伝送されるビデオデータからの別個の伝送は重要である。それは、ビデオ操作の処理の上のセッティングが効果的でない場合、システムのリアルタイム性能がかなり低下するからである。

#### 【0025】

本発明は、添付の図面に関する好ましい実施形態の以下の詳細な説明を読んだ後、よりよく理解されるであろう。

。

#### 【0026】

##### 【好ましい実施形態の詳細な説明】

上記の目的に合致し、本発明の現在好ましい典型的な実施形態に従った他の有

益な特徴を提供するシステム及び方法を、図1－8に関して下で説明する。当業者は、本明細書でそれらの図に関して与えられた説明は、単に説明的な目的あり、本発明の範囲を制限する意図はないことを容易に認識するであろう。従って、発明の範囲に関する全ての問題は、添付の請求項に言及することによって解決されなければならない。

#### 【0027】

##### 【システム概要】

図1に示すように、発明の実施形態のリアルタイムビデオ処理システム（VPS）は、多数のサブコンポーネントを有する4つの主要なシステム構成部品に分けられる。VPSの主要なコンポーネントは、

汎用マイクロプロセッサ又はデジタルシグナルプロセッサ（DSP）12を提供するプロセッサマザーボード（PM）10である。それは、専用のビデオハードウェアを制御し、容易にビデオハードウェアにマップされない像解析操作を実行し、及びVPSシステムの整数部分でない他のコンポーネントとの通信を促進する。

#### 【0028】

VPS内のベースラインビデオ処理コンポーネントでビデオ処理マザーボード（VPM）20である。各VPM 20は、始終一貫した速度（グローバルピクセルクロックに基づく）で、ビデオストリームの操作を実行することができる専用の並列パイプライン化ビデオハードウェアを含む。図1に示すように、専門イメージ取得、表示及び処理装置のためにビデオ処理ドータボードを（VPD）22－28と呼ばれるVPM20も1又は2のドータボードの加算をサポートする。図示のように、各々VPDのそれ自身のセットを有する単一のVPSシステム内に一以上のVPM20があることがありえる。もちろん、VPDは、本明細書で示され且つ説明されているそれらに限定されず、異なる構成で接続される他の処理ボードであるだろう。

#### 【0029】

A Global Video Bus（GVB）30は、専用の所有のビデオバックプレーンを定義し、ビデオ情報に伝達されことを可能にする33MByteでの一秒

につきVPSのVPM 20の間で、そのうえ、PM 10がビデオオンのマイクロプロセッサ12へ／から伝達されてある。

#### 【0030】

グローバル制御バス(GCB)40は制御バスをPM 10とVPM 20とVPSのGVB 30との間に提供する。PM 10からの制御レジスタアクセスは、VPS内の転送先ボードにGCB 40を越えて処理される。一般的に、ビデオランザクションは、このGCB 40を越えて実行されない。

#### 【0031】

三大VPSサブシステムの基本的操作可能及び構成上の見地を、今、説明する。次のセクションは、順番にVPSシステムの各コンポーネントの、より詳細な情報を提供する。「再構成可能(reconfigurable)」という用語は、本明細書では、2つの意味が与えられている点に留意する必要がある。即ち、新しい構造にプログラムすることができる制御レジスタを有する「再構成可能な」装置、及び異なるハードウェア関数を実行するために再構成されるFPGAのような「再構成可能な」装置である。

#### 【0032】

PM 10は、VPSのマイクロプロセッサコアとして機能する。2台のマイクロプロセッサ12は、ドータボードコンポーネントとして、更に1、2のマイクロプロセッサ12を加える可能性を有するPM 10で実際に使用される。PM 10の一次関数は、VPM 20及びその関連VPD 22-28によって実行されるビデオ処理操作のコマンドアンドコントロールを提供するようになっている。VPS内のビデオ処理操作は、ビデオハードウェアの制御レジスタの最初の準備を必要とし、操作実行の開始を定義する使用可能信号のアサーションが続く。PMのマイクロプロセッサ12は、レジスタプログラミング及びビデオ処理操作実行を実行するメモリーマップされたVPM/VPD制御レジスタにアクセスすることを通して。マイクロプロセッサ12のうちの1台以上にロードされるハイレベル、C-コール可能ハードウェア制御ライブラリは、ビデオハードウェアのコーディネーションを促進するために使用される。PM 10の第二の関数は、利用可能な専用のハードウェアを使用してより効率的に実行されることが

できない像の追加の処理を提供することになっている。そのGVB 30は、VPM 20の各々に各マイクロプロセッサのローカルメモリにビデオ情報を提供することを可能にする高速データバスを提供する。イメージ操作は、最終のアプリケーションによって必要に応じてイメージ上でマイクロプロセッサ12によって次に実行されることができる。PM 10の第3の機能は通信である。VPSがビデオインプット及びアウトプットで通常構成される一方、低い速度（より一般的なデータ通信）の必要性もある。そのような通信は、RS-232の連続的データ通信及びイーサネットを含むことができる。

### 【0033】

VPM 20は、専用のビデオ処理ボードである。VPS内の全てのビデオハードウェアは、並列パイプライン化方法によってビデオストリーム上で動作する。このビデオデータが、活性ビデオ情報をフレーム化する適切なタイミング信号で一度にフレームストアからピクセルを読込まれることを意味する。このビデオがシステムの中を流れると、VPM 20上の種々の処理装置によって処理され、又は、ビデオは転送先メモリでストアされることができる。VPM 20の全ての処理コンポーネントがデータ処理のためにフロースルー構成内で働くように設計される。各処理装置は、処理でパイプラインの遅れの固定した量を加えるが、システムのデータスループットを維持することが保証される。このように、ビデオフレームの上の操作を実行するために要求される時間は、常に決定的である、実行される操作に依存するパイプラインの遅れの固定した量を与えられる。システムを通してのビデオルーティングは、各VPM 20のデジタルクロスポイントスイッチの使用を通して実行される。このスイッチは、ソース位置からのビデオに、限定なしで、VPM 20の上のいかなる転送先にも発送されることを可能にする。また、クロスポイントスイッチは、ビデオがペナルティーなしで1つのソースから複数の転送先まで「散開する」ことを可能にする。クロスポイントスイッチルーティングを含む、全てのハードウェア操作は、VPM 20上でメモリーマップされた制御レジスタのプログラミングを通して定義される。各処理装置、クロスポイント接続及びストレージ装置は、特殊操作を定義するために操作される一組のレジスタを有する。PMのマイクロプロセッサ12は、これら



の制御レジスタをセッティングして、ビデオ操作が開始するのを可能にする役割を果たす。

#### 【0034】

V P D 22-28は、V P M 20上の装置と同様に機能する。各V P Dは、専用のクロスポイントスイッチインプット及びアウトプットを有し、メモリーマップされた制御レジスタを通して制御される。

#### 【0035】

更に詳細に下で説明するように、V P M 20内の幾つかのクロスポイントインプット及びアウトプットは、他のV P SボードへのV P M 20への、そしてそこからの、ビデオルーティング専用である。

#### 【0036】

G V B 30は、V P Sシステムボードの間でルーティングビデオに対してある役割を果たす。ビデオは、V P M 20とP M 10との間で、V P M 20の対の間で発送されることができる。アプリケーションに従って多数の異なる可能なG V B 30が存在する。最も単純なG V B 30は、固定トポロジーを有するV P Sシステムボードの間で専用の、ハードルーティングされたデータチャネルを提供する。これは、ボード間でビデオをルーティングするために、最も単純且つ最も経済的な方法であるが、それはまた、最も柔軟性が乏しい方法である。他に、G V B 30のより進んだバージョンは、アクティブなルーティング性能を含む。これらのルーティング性能は、二次のクロスポイントスイッチによって提供され、それはV P Sアクティブなバックプレーンの上で直接に存在する。このクロスポイントスイッチは、V P M 20のクロスポイントスイッチと同じ性能を可能にする、即ち、V P Sシステムボードからのいかなるデータアウトプットも、ペナルティーなしで1つ以上の転送先V P Sボードにルーティングされることができる。G V B 30を有するビデオルーティングの唯一の限定は、各V P Sボードの上で利用できるインプット及びアウトプットの数である。

#### 【0037】

そのG C B 40は、P M 10と他のV P Sシステムボードとの間でデータアクセスを実行する役割を果たす。P Mマイクロプロセッサ12での制御レジス

タアクセスは、常にGCB 40を使用して実行される。GCB 40は、VPSにバス内を流れるビデオから別個にしておかれ、存在するトラフィック（バンド幅及びレーテンシ）の量を最小にする。理論的には、GCB 40はマイクロプロセッサの大部分のタイプによって使用されるいかなる規格アドレス及びデータバスでもありうる。GCB 40の要件は、バス速度がかなり高いということである（典型的なランダムデータランザクションに対して250 nsec（8クロック）又はそれ以下のオーダー）。この速度は、多くの基準のバス（例えばVME）を使用して排除する。Compact PCI<sup>TM</sup>バスは、本発明の好ましい実施形態で使用される。このバスは、33MHzであり、VPSによって必要な適切な制御レジスタプログラミング速度を提供するフル32ビットデータバスである。

#### 【0038】

##### 【好ましい実施形態】

図1のVPSの主要な目的は、ビデオ処理を実行することであるので、初めにVPM 20、次に、PM 10及びそのハードウェア制御ライブラリ（HCL）、VPD 22-28、GVB 30、最後にGCB 40を説明する。

#### 【0039】

##### ビデオプロセッサモジュール

VPM 20のブロック線図及びGVB 30との相互接続を、図2で示す。VPM 20は、VPSの基本的ビデオ処理関数を提供する。上で言及したように、複数のVPM 20は、VPSに存在することができ、及び、各VPM 20は、専門機能のために最高2つのビデオ処理データボード（VPD）を含むことができる。図2で示されるVPM 20の好ましい実施形態で、各VPM 20は、以下のキーコンポーネントから成る。

#### 【0040】

39x39チャンネルノンブロッキングクロスポイントスイッチ202は、1チャンネルにつき10ビットであり、クロスポイントスイッチバスの上に伝達されるビデオデータの各ピクセルに対して2ビットのタイミング情報及び8ビットのビデオデータを表す。

#### 【0041】

4つの1K×2Kのピクセルフレームは、FS1-FS4(204-210)をストアする。これらのフレームストア204-210は、トリプルポートされ、全速ビデオ読み込み及び同時ビデオストアをできるようにする。第3のランダムアクセスポートは、また、フレームストアのビデオRAM(VRAM)のマイクロプロセッサダイレクトアクセスに提供される。

#### 【0042】

4つのピラミッド発生モジュール212-218。これらのピラミッドモジュール212-218は、前記の米国特許第5,359,674号及び米国特許出願連続番号第08/838096号において説明された、ポイントワイズイメージ変換用の関連8ビットルックアップテーブル(LUT)220-226を有するタイプのPYR-2つのフィルタリングASICを使用して実行される。16ビットデータの上でピラミッド操作を実行するために結合されることができるよう、ASICの各ペアは構成される。

#### 【0043】

1つの可変なALU(CALU)228。CALU 228は、一対のイメージの上で実行しているポイントワイズ操作のための方法を提供する。CALU 228は、自動タイミング整列そのインプットでタイミング補償板及びプログラムできるイメージ遅延230を含む、32ビットアキュムレータ及び16ビットアウトプット(16:16)ルックアップテーブル(LUT)への、16ビットインプットが続く

プログラムできるALU(PALU)232。PALU 232は、マルチイメージ操作に使用される。PALU 232は、DRAMの最高16MByteを有する再構成可能なフィールドプログラマブルゲートアレイ(FPGA)を含む。それは、4つのビデオインプット及び2つのビデオアウトプットをサポートする。PALU 232のより詳細な説明は、米国仮特許出願連続番号1997年9月8日に出願され、また、本譲り受け人に譲渡された第60/058,269号によって提供される。

#### 【0044】

2つのVPD部位234及び236は、異なるアプリケーションVPSを専門

化するためにデータボードコンポーネントをすえ付ける使用される。4つの異なるVPDを、更に詳細に下で説明する。好ましくは、VPM 20、GVB 30及び全てのVPDコンポーネントは、シングルシステムに広がるクロックに同期している。VPS内の既存の処理装置が与える達成可能な全速力で、このクロックは動作する。好ましい実施形態で、VPSのクロックスピードは、33MHzである。もちろん、このクロックは、速度が増加させられることができる半導体技術が、改善され、VPS上の最も遅いコンポーネントが、速度で同一基準において増加させられる。VPM 20は、VPM 20、ビデオストレージモジュール、GVB 30及びPM 10の間でビデオ移送標準化されたビデオフォーマットを使用する。このビデオフォーマットは、ピクセルにつき8ビットのデータを含み、更に水平(HA)及び垂直のアクティブな(VA)データの領域を示すことによってアクティブなビデオデータをフレーム化する2つのタイミング信号を含む。イメージ内に各アクティブラインの間の固定したブランキング間隔存在する。このブランキング期間は、デアサート(deasserted) (ロー) HA及びアサート (ハイ) VAで定義される。各イメージの開始及び終了でのブランキング期間もあるだろう。全てのビデオデータは、VPSシステムクロックと同期している。故に、ビデオデータは、システムを通して33Mbyte/secの速度又は等しく33Mpixel/sec速度で伝送される。これらの速度は、イメージ内の存在するブランキング時間を含まない。全ての処理、ルーティング及びVPSでのストレージコンポーネントは、このフォーマットでのビデオを受け入れ、同じフォーマットでビデオアウトプットを提供するように設計される。

#### 【0045】

本発明の実施形態に従うと、ビデオデータは、幾つかの理由でHA及びVAタイミングでフレーム化される。その理由とは、

1) パイプラインの遅れの補償作用。全てのVPS処理装置及び大部分のルーティング装置は、ビデオ信号でパイプラインの遅れを負う。HA及びVAタイミング信号の使用を通して、各ビデオイメージが受けるパイプラインの遅れの量は、プログラムすることによって重要ではない。それは、ビデオ信号のタイミングがタイミング情報を通して既知であるからである。

**【0046】**

2) タイミング情報も、H A及びV A信号の解析を通して自動的なパイプラインの遅れ同期を可能にする。例えば、ハードウェアの2以上のイメージを結合するとき、最小の遅延を有するビデオデータストリームは、それが結合される他のビデオデータストリームと同じタイミングを有するように遅れなければならない。このタイミング同期は、パイプラインの遅れアプリアリを知ることなく自動的に実行されることができ、そしてそれは、システムのプログラマ多くのハードウェア制御機能を非常に単純化する。

**【0047】**

3) イメージサイズの知識。H A及びV Aタイミング信号は、ビデオ装置がアプリアリ情報なしでイメージの真のサイズが知ることができるようにする。一例として、フレームストアは、イメージサイズのアプリアリ知識なしで適切に正しいイメージサイズのイメージをストアすることができる。これはまた、例えばピラミッド処理A S I C 212-218のインプットライン遅延及びボータ制御等のビデオ処理の多くの他の見地を単純化する。

**【0048】**

4) ビデオルーティングチャネルの独立。H A及びV Aタイミングがない場合、V P Sは、イメージサイズとパイプラインの遅れアプリアリの両方を知らなければならない。これはビデオ操作の解析の過大な量を必要とし、ビデオ装置のハードウェア制御の、複雑化を引き起こす。タイミングチャネルが、システム内の幾つかのビデオバスに提供され、他のビデオバスと共有される場合、V P Sの中を流れることができる独立のデータバスの数に限界がある。

**【0049】**

要するに、別個のH A及びV A信号を有する、各ビデオバスをフレーム化することを通して、V P Sの全ての装置を同時に通して独立のタイミングを有する独立のビデオ情報をルーティングすることは可能である。このように、本発明の実施形態のV P Sは、全体として並列ノンブロッキングルーティング及び処理を提供する。

**【0050】**

V P S内の並列パイプライン化されたハードウェアは、データを有する信号をフレーム化するビデオを使用して、ビデオデータのアクティブイメージ及びブランキングの領域を描写する。この情報は、上で言及したように、V P S内のビデオ装置を単純化することに役立ち、ソフトウェア化制御を通して容易にプログラムできるそれらの装置を作る。V P Sの重要な他の見地は、ビデオ処理操作同期開始を提供することである。複数のビデオパスビデオタイミングが、ビデオタイミングが、既知の初期条件で開始することを確保するために同期方法で開始することは、避けられない。この保証がないと、ビデオハードウェアは、複数のストリームでに操作を実行しているときに、常にタイミング補償作用を実行しなければならない。それは、多重チャネルからのタイミングの最初の差異（またチャネルの間のタイミングスキューとして知られている）が未知となるからである。R D \_\_\_\_ S T A R Tと呼ばれる一般の制御信号がV P S内に提供されるソースビデオ装置からのビデオ読み込み操作のための同期開始を提供する。R D \_\_\_\_ S T A R Tがアサートされるとき（マイクロプロセッサコントロール下のレジスタへの書込みを通して）、全ての前に可能にしたビデオソース装置は、同期態様で外に読み出し始める。これは、後続の処理のビデオタイミング解析を単純化するのに必要なビデオタイミングのための既知の初期条件でプログラマを提供する。好ましい実施形態で、R D \_\_\_\_ S T A R T信号は、指定された「主な」V P M 20の上で発生させられ、（複数のV P M 20を有するシステムで）V P Sで全てのV P D及びV P M「マスタ」を含む全てのV P M 20によって同時に受け取られる。「マスタ」V P M 20の使用は、V P Sが、独立の複数のR D \_\_\_\_ S T A R Tを有する複数の「マスタ」を有することを制限しない。各R D \_\_\_\_ S T A R Tは、選択されたR D \_\_\_\_ S T A R Tマルチプレクサを通して異なるソースから制御されるだろう。複数のR D \_\_\_\_ S T A R Tは、非同期独立のビデオ操作が実行されることができるようになる。

#### 【0051】

好ましくは、V P Mクロスポイントスイッチ202は、ビデオ装置のアウトプットに存在するビデオを、更なる処理又はストレージするための他のビデオ装置のインプットに提供する。上で言及したように、V P Mクロスポイントスイッチ

202は、好ましくは10ビットの各チャネルデータを有する39x39クロスポイントスイッチとして実行される。これらの10ビットは、8ビットのデータ及び2ビットのHA及びVAタイミング（上記のフォーマットの）を表す。好ましい実施形態で、クロスポイントスイッチ202は10のI-CUBE 1Q96クロスポイントスイッチ装置を使用して実行される。これらの装置の各々は、ビデオデータの10ビットの単一のビットのための、再構成可能なクロスポイントスイッチを提供する。1つの制御レジスタ書込みが同時に全ての10のIQ96装置でデータの全ての10のビットを切り換えることができるように、これらの装置は構成される。

#### 【0052】

クロスポイントステートマシン238が、クロスポイントスイッチ202で各インプット及びアウトプットデータバスと関連しており、それは、バス上にビデオ移送をモニタする。クロスポイントスイッチ202の全てのビデオインプット及びアウトプット信号のビデオアクティブ（VA）データに応じて、クロスポイントステートマシン238は、バス上に伝達されるビデオについてのステータス情報を提供する。クロスポイントステートマシン238を使用して、クロスポイントスイッチデータバスが現在使用されているかどうかは、又はそれがアイドルかどうか決定することが可能である。アウトプットポート（XB\_\_OUT）及びインプットポート（XB\_\_IN）のクロスポイントステートマシン238の4状態は、図2A及び2Bで示されて次のように説明される：

IDLE. これは、現在データ流れがデータバスを通っていないことを意味し、データバスは、データのルーティングのために割込み可能でなかった。クロスポイントスイッチ202がIDLE状態であり、VAが検出されるならば、状態はERROR状態に変化する。

#### 【0053】

ENABLED. ルーティング操作を実行するようにデータバスがプログラムされるときステートマシン238は、アイドルからイネーブル状態に変化する。VAは、この状態の間検出され、BUSY状態への状態変化する。

#### 【0054】

BUSYは、データバスが故意にルーティングされたのち現在このデータバスを通るアクティブなビデオ流れがあることを指示する。VAの立下りエッジが検出されるとき、ステートマシン238はIDLE状態に変化する。

#### 【0055】

ERROR、エラー状態は、プログラマがそのバスを通してビデオを故意にルーティングしなかったが、VAがデータバス内の検出されたことを示す。この状態は、クロスポイントバスを所定のインプット（又はアウトプット）バスに接続することによって、単にステートマシン238を明示的にクリアすることを通してクリアされることができる。

#### 【0056】

クロスポイントステートマシン238は、2つの重要な理由に対して決定的である。最初に、GVB30はクロスポイントチャネルが現在利用されている及び利用できる知識に依存する。これは、プロセッサ12のハードウェア制御ライブラリの内部にグローバル状態情報を保つことなく、各VPM20への、そして、そこからの、ビデオバスのかなり単純且つかなり高速なアロケーションを可能にする。第二に、ビデオ操作が完了された後、自動的にクロスポイントアウトプットをターンオフすることは、プロセッサ12のハードウェア制御ライブラリにとって困難である。それらが使用された後アウトプットバスをターンオフすることは、追加の管理作業必要とし、各ビデオ操作中のに書込まれた制御レジスタの数を増加させる。クロスポイントアウトプットが、ターンオフされないとき、無関係なビデオ移送からの不必要なビデオは、現在使っていないアウトプットに提供されることができる。特に、ソース装置がある操作に使用されるとき起こり、異なる転送先装置を有する後続の操作に使用される。第一の転送先装置は、まだソースのアウトプットに接続しており、第二の搬送から偶然にビデオを受け取る。幾つかのVPS装置、特に、ピラミッド処理ASIC212-218は、ビデオの意図的でない受取りによって処理の望まれていない状態に設定されることができる。クロスポイントスイッチ202が与えられたバスERRORモードである場合をモニタすることを通して、プロセッサ12のハードウェア制御ライブラリによって、この状況は検出されることができ、改善アクションがとられる



ことができる。

#### 【0057】

上で言及したように、VPMフレームストア204-210は、トリプルポートのアクセス性能を有する1K×2Kピクセルフレームストアとして編成される。フレームストアは、フルビデオレート（33MHz）で、ビデオ読取り及びビデオストア操作を実行することができるようになっている、GCB 30からのマイクロプロセッサデータバスを通してフレームストアメモリのランダムアクセスに利用できる第3のポートを有する。好ましい実施形態で、フレームストア204-210は、メモリ（204-210）及びフレームストアコントローラ240として、Xilinx 4013のフィールドプログラマブルゲートアレイ（FPGA）で各々実行される。フレームストアメモリ204-210は、標準1K×512のピクセルビデオRAM（VRAM）を使用してそれ自身実行される。しかし、ビデオ操作に関して、フレームストア204-210は1K×2Kピクセル2Dフレームストアとして編成される。このように、イメージ読取り及びストア操作は、全ての方形関心領域（region-of-interest（ROI））で実行され、それは、フレームストアメモリ204-210の大きさ内に存在するであろう。この複数イメージフレームストア内に容易に保存されることを可能にし、また、フルイメージの小区域にアクセスすることを可能にする。

#### 【0058】

読取り操作中、局所的な1VPM制御バス242に接続し、インプットビデオデータを通るフレームストア制御装置240は、様々な異なるモードをサポートするためにフレームストア204-210を制御する。ビデオ読取りは、最短隣接ピクセル拡張のプログラムできる量で実行されることができ、このように、オリジナルイメージのズームされたバージョンであるアウトプットイメージを提供する。また、ビデオ読取りは、インタレースモードで実行されることができ、フレームストアイメージで一行毎に読込むメカニズムを提供する。インタレースモードは、インタレースビデオを扱う場合特にフレームストア204-210がビデオディスプレイとして使用される場合大変有用である。全てのビデオ読取りモードは、フル二次元のROI操作をサポートする。

## 【0059】

本発明の好ましい実施形態で、ビデオストア操作は、また、イメージのリサンプリングで実行されることができる。イメージストア中のイメージは、プログラムできる量によって10分の1を取られることができる（サブサンプリングされる）。サブサンプリングが使用されるとき、ピクセルのサブサンプリングのフェーズ（第一、第二、第三などであることを示す）はサブサンプリング中に保存される第一のピクセルである。また、ビデオ読取り操作のように、ビデオストア操作は同様にインタレース記憶操作を提供する。一組のコントロールレジスタのプログラミングを通して、フレームストア制御装置240内でビデオ読取り及び記憶操作は定義される。VPS内の前記のRD\_\_START信号は、読取り操作の開始を合図する、全てのプログラムされる及び割込み可能なフレームストアから同期読み出しを提供すること。

## 【0060】

しかし、フレームストアコントローラ240は、好ましくはビデオ操作の別個の、非同期読取り及び記憶クロックを使用する性能を有する。VPM 20のフレームストア204-210はグローバルVPSシステムクロックに同期した操作の全ての読み込み及び保存を実行するが、フレームストアコントローラ240は、異なる周波数の読取り及び記憶クロックを使用することができる。また、フレームストア204-210は、連続的読取りモードで動作することができる、ビデオは、内部発生ビデオタイミング及びRD\_\_START信号を使用することを通すのではなく外部タイミング信号をベースに読込まれる。読込み分離、及び記憶クロック及び連続的読み込み操作は、フレームストア204-210が両方とも汎用フレームストアとして（VPM 20で使用されているように）、また表示アウトプット用ビデオエンコーダを駆動しているフレームストアとしても使用されることができるようにする。

## 【0061】

上で言及したように、ピラミッド処理ASIC 212-218は、米国特許第5,359,674号及び米国特許出願連続番号08/838096において説明されるタイプのPYR-2ASIC（「ピラミッドチップ」として知られ

ーノフ株式会社によって製造された)を使用して、好ましくは実行され、PYR-2チップのOUT2に各々連結された8ビットLUT 220、222、224又は226を有している。そのピラミッドモジュール212-218のねらいはビデオフレームの上で基本波フィルタリング演算を提供することである。ピラミッドモジュール212-218のアウトプットでの8ビットLUT 220-226は、単一のイメージポイントワイズ操作がピラミッドモジュール212-218からのイメージアウトプットの上で実行されることを可能にする。そのような操作はゲインをイメージに加えること、イメージをスレッシュホールドすること等を含む。LUT 220-226は、32K×8のSRAMとして好ましくは実行され、カスタマイズされたLUT変換に完全にプログラムできる。

#### 【0062】

当業者は、ピラミッドモジュール212-218及び、フレームストア204-210の結合は、多種多様な異なるイメージピラミッド及び他のマルチ解像度イメージ表現がVPMビデオハードウェア内で完全に実行されることを可能にすることを認識するであろう。可能なピラミッド計算の実施例は、米国特許第5,359,674号及び米国特許出願連続番号08/838096において提供される。

#### 【0063】

CALU 228は、そのインプット及び512K×16バンクに2つのFIFOバッファ230を有するCALUコントローラとしてXilinx XC4010 FPGA又はXilinx XC4028を使用して実行される。そのCALU 228のねらいは、2つのイメージの上で任意、ポイントワイズ操作を使用して2つのイメージ操作を提供することである。インプットFIFO 230は、両方とも自動的に、2つのインプットビデオバス間のいかなるタイミングスキューも補償し且つ、2つのイメージの間で決定的な、プログラムできる遅延を提供するCALU 228の制御レジスタを通し、プログラムできる。

#### 【0064】

2つのインプットビデオチャネルの間の自動タイミングスキュー補償作用は、2つのインプットビデオチャネルのタイミング信号を分析することによって達成

される。第一の妥当ビデオデータを提供するインプットビデオチャンネルは、妥当ビデオデータが第二のビデオチャンネル（F I F Oのサイズによって単に制限されて）に現れるまで遅れる。この特定のインプリメンテーションで、両方のビデオチャンネルからのビデオデータは、データF I F O 230に保存される。両方のF I F O 23に十分な有効なデータが保存されるとき、そしてF I F O 230のデータアウトの読み込みプロセスは両方のチャンネルの上で同時に始まる。各チャンネルは、プログラムできる遅延によってF I F O 230からデータを読込むために次に開始を遅らせることができる。

#### 【0065】

ビデオインプットチャンネル上で、ビデオチャンネルインプットのタイミング解析セクションは、アクティブな及びブランキングクロックの数を検出し、同じタイミングパラメーターを有するアウトプットビデオストリームを構成するためにこれらの数を使用する。

#### 【0066】

自動タイミング補償作用は、また、P A L U 232で実行されることができ、しかし、チップに外部の明確なメモリコンポーネント（F I F O）を有しないのでそれは、内部リソースを使用する必要がある。このために、幾つかのアプリケーションは、P A L Uでタイミング補償作用を含まないように選択し、又はより小さいタイミング差を補償するだろう。

#### 【0067】

C A L U 228は、16ビットインプット及び16ビットアウトプットL U Tを通してポイントワイズイメージ操作を実行し、2インプットピクセル値に基づく独特な16ビットのアウトプット値を発生させる。そのL U Tは、S R A Mで実行されて及ソフトウェアを通してプログラムできる。一般の操作、例えば、イメージ乗算、加算、その他は、既知のやり方でこれらのL U Tを使用して実行されることができる。より複雑な操作（あるイメージの水平及び垂直の偏導関数に基づく勾配データの角及び大きさを発生させることのような）また、彼らのプログラム可能な性質のためにC A L U L U Tで可能である。実際、変換がインプット値の各独特なペアの単一のアウトプット値を発生させる場合いかなる2つ

からなるイメージ操作も、CALU LUTで実行されることができる。

【0068】

CALU 228も、内部で32ビットイメージアキュムレータを有する。このアキュムレータは、インプットイメージのうちの1つ又はCALU LUTのアウトプットを、イメージの全体の大きさの上に蓄積されることができるようにする。このイメージの平均値を決定する速い方法を可能にし、及び、また、フルイメージ交差相関計算のような操作に使用されることができる。

【0069】

好ましくは、CALU 228（本発明の実施形態のVPSでの全ての他のALU及びFPGAと同様に）は、再構成がJTAGインタフェースを通してPM 10のマイクロプロセッサ12のうちの1台以上のソフトウェアから、あるところで異なるハードウェア機能に対して再構成可能である。

【0070】

PALU 232は、多数の異なるビデオ操作再構成可能な装置として設計された。PALU 232は、JTAGコントロールを通してプログラムされるように設計され、連続的通信チャネルはパワーオンリセットの後のPM 10のマイクロプロセッサ12によって境界線スキャンを使用してテストデバイス設計される。PALU 232は4つのビデオインプット及び2つのビデオアウトプットを有し、VPSに16ビットの2個一組の機能性能を提供する。PALU 232は、それに接続している4M×32DRAMを有し、それは、大きい、32ビットワイドメモリバンクを使用することができるソフトウェア配列を通して多種多様な処理関数が実行されることができるようになっている。PALU 232は、装置を構成するために使用される構成データに従ってこのように異なるビデオ操作のホストを実行するようにプログラムされることができる。PALU 232の構成及び操作は1997年9月8日に出願され、本出願の譲り受け人に譲渡された、米国仮特許出願連続番号第60/058,269号で詳述されている。

【0071】

VPM 20のVPD部位234及び236は、VPSの性能をカスタマイズ

及び拡大するために提供されている。ビデオディスプレイ、ビデオデジタイザ、相関性ユニット、イメージワーパ及び他の処理装置のような専門化されたビデオ装置は、ドータボードデザインに取り入れられることができV P Sに追加される。各V P Dサイト2 3 4及び2 3 6は、どのV P Dがすえ付けられるかによってV P Mのクロスポイントスイッチ2 0 2への、そして、そこからの、ビデオのための、最高6つのクロスポイントインプット及び6つのクロスポイントアウトプットを有する。また、各V P Dは、インタラプト駆動ハードウェア制御を調整するためにそれと関連する4つのインタラプトを有する。

#### 【0 0 7 2】

上で言及したように、重要な関心事項である、発明の実施形態のV P Sのデザインは、単一の又はマルチプロセッサによるハードウェアの効果的なコントロールであった。ビデオフォーマット、R D \_\_\_\_ S T A R T信号の使用及びC A L U 2 2 8の自動タイミング補償作用の全てが、V P Sを容易且つ効率的にソフトウェアでプログラムすることを可能にする。V P Sがマルチタスキング、マルチプロセッサ環境内で制御されるためには、V P Mビデオ装置は、インタラプト駆動コントロールを通して制御されなければならない。タスクがビデオ操作の完了を待っている一方、インタラプトがスイッチングタスク及びブロッキングタスクのための方法を提供する。インタラプトなしで、タスクはV P Mビデオデバイスステートレジスタ内の、ステータスピットをポーリングし、操作が完成するときを決定しなくてはならない。この速いタスクスイッチング及び効果的な操作にそれ自体に良好に役に立たない。

#### 【0 0 7 3】

V P Sでの重要なインタラプトは、ビデオ操作の完工を合図するインタラプトである。他の方法で説明した、ビデオシンクとして役に立つ装置によって、インタラプトは発生させられなければならない：ビデオは装置に入るが、装置を残さない。インタラプト発生にとって重要であるV P M 2 0の装置及び操作は、以下の通りである

フレームストア2 0 4-2 1 0の記憶操作の完工。

#### 【0 0 7 4】

CALU 228内の操作の完工。

【0075】

PALU 232内の操作の完工。

【0076】

VPDの関連する操作の完工4つのインタラプトが各VPDに割り当てられるならば、VPM 20の上で発生させられることができるインタラプトの合計数は14である（フレームストア4、CALU 1、PALU 1及びVPD 8）。最高6つのVPM 20が好ましい実施形態でVPS内の存在することができるように、非常に多くの割込み信号が、インタラプト発生のためにPM 10にバックプレーンを横切って独立して実行される。インタラプト発生の問題を低減するために、割込みマスク及びフラグレジスタ244は、各VPM 20の上で提供され、単に1つのインタラプトが、VPSで各VPM 20からPM 10まで提供される。VPM 20の上でビデオ装置によって発生させられるインタラプトは、関連マスクビットが装置のために割込み可能な場合だけPM 10に伝達される。マスク及びフラグレジスタ244は、PM 10のマイクロプロセッサ12どれかによって、マスク及びフラグステータスビットが原子的にセット又はクリアされるように設計され、マルチタスキング及びマルチプロセッサ制御操作中に問題が発生しないことが確保される。

【0077】

好ましくは、ビデオ同期信号は、全てのVPM 20及びVPDを横切って分散されるそれで、外部のビデオへのインタフェースが合図するビデオ装置の全ての部分は、お互いに同期することができる。特に、システム表示は、ビデオデジタイザに通常同期する。この機能は、幾つかの外部ビデオソース間で決定的なタイミングソースを提供し、フレーム率処理にとってかなり重要である。

【0078】

VPM 20のフレームストアメモリ、制御レジスタ及びLUTメモリは、GCB 40を通してPM 10上のマイクロプロセッサ12でアクセスされることができる。上で言及されるように、GCB 40はCompact PCI<sup>TM</sup>バスとして実行され、各VPM 20はPCIスレイブ（GCB）コントローラ246を

有し、該コントローラはG C B 40からP C Iアクセスを復号し、V P M 20内に存在する内部ローカルな制御バス242を経てV P M 20上の装置へアクセスを進める。このように、各V P M 20は、制御レジスタ及びメモリアクセスそれ自身のローカルな制御バス242を有する。

#### 【0079】

G C Bコントローラ246は、また、J T A Gドライバとして機能し、V P M 20のF P G A及びいかなるドータボードにも構成データを提供する。特に、G C Bコントローラ246は、V P M 20の上のF P G Aの再構成のためにJ T A Gコントロールを提供するだろう（例えばV P DのいかなるF P G Aとも同様に、C A L U 228、P A L U 232、フレームストアコントローラ240、クロスポイントステートマシン238及び割込み制御器244） 最終的に、本発明の実施形態のV P Sが非常にモジュラであるので、V P Sの電流構成がソフトウェアから決定されることができることは、重要である。Compact P C I<sup>7</sup>（それはG C B 40使用される）は、どのスロットが占められるか及びそれによる機能について確認するためにプロトコルを取り入れる。このプロトコルは、本発明の実施形態のV P Sの、P M 10及びV P M 20のボードの上で好ましくは実行される。加えて、各P M及びV P Mボードは、好ましくはリビジョン及びハードウェア構成情報を組み込み、V P Dのレベルのタイプ及びリビジョンを識別するボードコネクタの一部として各V P Dが好ましくは物理的に組み込まれた(hardwired)コードを組み込む。このコードは、V P M 20上で特殊アドレス位置から読込まれることができる。一旦V P Dタイプが確立されるならば、システムソフトウェアは、例えばV P Dハードウェアコンフィギュレーション等のV P Dのより詳細な情報を得るために、V P D 20上のどの基本的レジスタにアクセスするかを知っている。このように、システムマスタは、V P Sを通して適切なレジスタに問い合わせることができ、ソフトウェアで全てのビデオ装置を識別して及びカタログを作る。

#### 【0080】

##### プロセッサマザーボード

プロセッサマザーボード(P M) 10は2つの主要なサブシステムヲ含む。各



マイクロプロセッササブシステム12がアクセスすることができるマイクロプロセッササブシステム12及び共用可能装置である。PM'のマイクロプロセッサ12は、本明細書で一般にマイクロプロセッサとして説明された。しかし、好ましい実施形態で、PM10は、その汎用マイクロプロセッサとして専門化されたデジタルシグナルプロセッサ(DSP)12を使用する。

#### 【0081】

PM10のトップレベルのブロック線図を、図3で示す。図示のように、トップレベルのPM構成は4つのマイクロプロセッサブロック302-308を含む。好ましい実施形態で、各マイクロプロセッサブロック302-308は、60MHzで独立して実行するテキサスインスツルメンツTMS320C40DSPを含み、各マイクロプロセッサ302-308は、ローカルSRAMの3つのバンクを有する。即ち、データSRAM領域310-316、プログラムSRAM領域318-324及びビデオSRAM領域(VSRAM)326-332。これらのローカルなSRAMの各々は、単にそのローカルなバスを介してローカルなマイクロプロセッサ302-308でアクセスできる。加えて、PM10は任意の制御バス334を通してマイクロプロセッサの全てでアクセスできる一組の共用リソースを有し、PMCB(PM Control Bus)と呼ばれている。そのPMCB334は、ラウンドロビンの方法でマイクロプロセッサ302-308の各々から任意にトランザクションする、このように、全てのマイクロプロセッサにソフトウェア排除又はソフトウェアアービトレーションなしでこれらのリソースにアクセスするのを可能にする。

#### 【0082】

好ましい実施形態で、共用リソースであるPM10の装置は、グローバルデータストレージのための4M×32DRAMメモリ領域336までを含む。マイクロプロセッサ間で大容量のデータを移送するための大きく伝達するメモリリソースがとして、DRAM領域は使用され、同一の位置で各マイクロプロセッサのアドレスにメモリマップされる。共用リソースも、使用される不揮発性の4M×16フラッシュEPROMメモリ338を含み、それはプログラム及びデータストレージ埋め込まれたスタートアップ及び実行に使用される。フラッシュメモリ

338は、埋め込まれた操作のために新しいデータ及びプログラムを実行可能形式に「バーニング(burning)」するための全てのマイクロプロセッサによって再プログラム可能である。他方、PM 10への外部の通信は、4つの共有RS-232のチャンネルを通して提供される。2つのDUART 340が、PMCB 334に提供される、UARTにアクセスするマイクロプロセッサ12が、4つのチャンネルの何れに対してもがデータを送受信することができるようにする。また、全てのマイクロプロセッサ302-308に対するJTAGエミュレータポート341のアクセスは、PMCB 334を通して提供されるだろう。更なるPMCB 334は、特別に設計されたハードウェア相互排除セマフォレジスタ343へのアクセスを提供し、また、インタラプトフラグ及びレジスタ、関連VS RAMコントローラ、及びマイクロプロセッサのローカルバス及びPMCB 334を接続するインタフェースを含むマイクロプロセッサブリッジ342-348へのアクセスも提供する。PCIブリッジチップ350は、PMCB 334とGCB 40との間にインタフェースを提供し、GCB 40 (Compact PCI<sup>TM</sup>規格で定義されるように)のマスタコントローラとして機能し、一方、PCIアービタ352はGlobal Control Bus 40のCompact PCI<sup>TM</sup>規格で説明されたようにPCIバックプレーン上の複数のPCI装置のアービトレーションを扱う。

### 【0083】

図3で示すように、全ての共用リソースに加えて、PM 10は、実際のマザーボード上で全ての4つのビデオSRAM (VS RAM) メモリ領域及び2台のマイクロプロセッサを有する。2つのプロセッサデータボードサイト354及び356は、第3の及び第4のマイクロプロセッサ306及び308を加えるために存在する。プロセッサデータボードサイト354及び356は、同一であり、また関連するプログラム及びデータSRAMを有する。

### 【0084】

好ましくは単一のC40 DSP 306又は308を含む。PM 10へのプロセッサデータボードサイト354又は356からのコネクタは、全ての関連するマイクロプロセッサバス及び制御信号をPM 10に接続する。このように

、PM 10はその構成内に4つのマイクロプロセッサブロック302-308を有する。マイクロプロセッサブロック302及び304のうちの2つは、PM 10の上に存在し、他の2台のマイクロプロセッサブロック306及び308は任意であり、PM 10のドータボードに設置することができる。好ましくは、各マイクロプロセッサブロックは、C40 DSPを含み、各DSPは独立の60MHzクロックから動作し、PM 10のグローバル資源はGCB 40と同様に33MHzクロックから動作する。マイクロプロセッサは、その2台の外部のデータバスに接続している128K×32又は512K×32のSRAMバンク（例えば310、318）を有する。これらの2つのSRAMバンクの目的は、ローカルマイクロプロセッサ処理に必要な、プログラムストレージ領域及びデータの、ローカルな、ゼロの待ち状態メモリを提供することである。PM 10の各マイクロプロセッサは、マイクロプロセッサ302のメモリマップが追加の装置を有すること以外は、同一のメモリマップを有する。ローカル資源は各マイクロプロセッサにおいて同じこと領域にマップされ、共用リソースはまた、同じアドレスにマップされる。加えて、全てのVPMコンポーネントは、また、同じアドレスにマップされる。これは、異なるマイクロプロセッサでの実行可能プログラムのための、異なるメモリマップ及び異なるリンキングコマンドのいかなる要件も除去する。各マイクロプロセッサも、ビデオSRAM又はVSRAM（例えば326）と呼ばれる第3のSRAM領域を有する。そのVSRAMは、512K×16SRAM領域として編成される。VSRAMは、インタフェース回路（図示せず）を介してGVB 30に連結されており、及び、VPM 20及びGVB 30によって使用される同じフォーマットのビデオ情報を送り出し又は受け取ることができる。2つのイメージデータバスは、GVB 30から各VSRAMまで接続しており、それによって、GVB 30ソースからVSRAMに直接に、最高66MByte/secのデータ転送を提供する。

#### 【0085】

各VSRAMの目的は、効果的な態様でVPM 20の上のビデオ装置からマイクロプロセッサ12までビデオ情報を提供することである。VSRAMでのストレージの後、マイクロプロセッサ12は、VSRAMのイメージ情報上で動作

することができるPM 10上のいかなる共用リソースも利用することなくVSRAMのイメージ情報上で動作することができる。このように、PM 10の4台のマイクロプロセッサ12の各々は、互いに妨害することなくそれらのVSRAMを使用し、又は、互いのリソースを利用してイメージ情報を受け取り、送り出し又は処理することができる。VSRAMを有するマイクロプロセッサのこの効果的な搬送及び解析性能は、リアルタイム環境での高速ビデオ操作に優れている。

#### 【0086】

当業者は、VSRAMは、16ビットのデータエリアとして定義され、C40構成のために幾つかの困難を提供する、そしてそれは、本質的に単に32ビット語幅であることに注意を要する。異なるメモリーマップ領域を介して、VSRAMにアクセスすることを通して、VSRAMからの読取りはが実行されることができ、VSRAMでの値又はフル16ビットの整数（サインエクステンションを有する及び有しない両方びもの）の単に最下位又は再上位バイトにアクセスする。これは、VSRAMデータを判断するソフトウェアの部分の追加の考慮事項なしで、サインされ及びサインされなかったデータを有する異なるワード幅のデータを扱うかなり効果的な方法を提供する。

#### 【0087】

図3で示さないが、当業者は、また、TI C40 DSP構成がプロセッサ通信のためにプロセッサに6つの非同期通信ポートを提供すると認識するであろう。各々、これらの非同期式ポートのうちの2つを使用して、PM 10の4台のマイクロプロセッサ12は互いに接続されている。このように、各マイクロプロセッサ12は、送出し及び受取りポートを、VPSのマイクロプロセッサ毎に有する。これは、マイクロプロセッサ性能への影響が最小で、PM 10の上でグローバル資源を利用しないマイクロプロセッサの間で、データの速い、効果的なDMA移送を提供する。各C40 DSPは、5つの外部のインタラプトを有する。これらの5つのインタラプトの各々は、ブリッジ342-348のインタラプトフラグ及びインタラプトマスクレジスタに接続しておりそれは、マイクロプロセッサ12に局在している。これらのフラグ及びマスクレジスタは、UAR

T 340、PCIブリッジ350、VPM 20及び他のマイクロプロセッサ12から、直接にマイクロプロセッサに接続されるインタラプトまで、複数の外部のインタラプトを多重送信するために使用される。これは、各マイクロプロセッサ12に他のPM装置から及びVPM装置からの多種多様な割込み要求に応答するメカニズムを提供する。

#### 【0088】

PM 10の好ましい実施形態で、2つのハードウェアセマフォレジスタ343が、あるPMCB 334上の共有資源として利用でき、マイクロプロセッサ12上で複数タスクによってハードウェア制御の実行及び調整を促進するように特別に設計されている。これらのセマフォは、以下の動作を示す。

#### 【0089】

セマフォレジスタ値が0であるとき、セマフォレジスタに書込まれるデータ値は、セマフォレジスタ343に格納される。

#### 【0090】

セマフォレジスタ値がゼロ以外のとき、セマフォレジスタに書込まれるデータ値は、セマフォレジスタ343の値に影響を及ぼさない。

#### 【0091】

0値がセマフォレジスタに書かれる場合、セマフォレジスタ343は、0にセットされる。

#### 【0092】

このセマフォレジスタ343は、オーバヘッドの過大な量なしで多重タスク及びマルチプロセッサ制御相互排除を実行するための手段を提供する。VPSシステム（最も重大なものは、RD\_\_START）での一定の重要な制御信号及びレジスタは、一度に1つのタスクによってのみアクセスされなければならない。このセマフォレジスタ343を使用することを通して、タスクがセマフォレジスタ343に書く独特な値があるときは、セマフォレジスタ343から読み出される時、タスクは相互排除セマフォを「授けられる(granted)」。タスクが操作で仕上げられるとき、タスクはセマフォレジスタ343に0値を書き、効果的にセマフォ及び他のタスクが「自由にし」アクセスすることができるようにする。こ

のセマフォレジスタ343なしで、複雑なインタラプト及びマルチプロセッサ制御共同は実行される必要があり、それはPMのマイクロプロセッサ12が、ハードウェア操作を実行するためのリアルタイム処理の可能性を低減するであろう。相互排除セマフォの使用は、重要なVPSリソースに対して、タスクからタスク及びマイクロプロセッサからマイクロプロセッサの相互排除の効果的な手段を提供する。

### 【0093】

図3で示すように、実施形態でのマイクロプロセッサ302、PM 10上の4台のマイクロプロセッサ12のうちの1台は、他のマイクロプロセッサ12に利用できない関連特殊ローカル資源を有する。これらのローカル資源のうちの1つは、ソフトウェアのパワーオンブーティングブーツEPROM 358である。他のマイクロプロセッサ12はリセットモードで保持され、通信ポートをわたるブーティング情報を待ち、マイクロプロセッサ302は、自動的にローカルブーティングEPROM 358からブートする。この埋め込まれたソフトウェアが、製造の後PM 10の上に配置されることを可能にし、常に、PM 10がソフトウェアのベースラインセットで最初にブートアップされることを確保する。一旦マイクロプロセッサ302の最初のブーティングが完成しているならば、新しいプログラムは、ロードされることができ、フラッシュメモリ338の外でマイクロプロセッサ302で実行されることができ、マイクロプロセッサ302はまた、他のマイクロプロセッサが同様にブートするように指示することができる。

### 【0094】

マイクロプロセッサ302だけに利用できる他のローカル資源は、10BaseTイーサネットコントローラ360である。コントローラ360は、マイクロプロセッサ302がイーサネット通信を実行することを可能にする。イーサネットコントローラ360は、イーサネット通信リンク上に提供されることができ、データの潜在的に高い量のため、PMCB 334によってサポートされる共用リソース領域よりもむしろ、マイクロプロセッサ302のローカルバスに配置される。このイーサネットコントローラ360は、AMD C-LANCEイーサネット装置を使用して好ましくは実行され、イーサネットパケットストレージ用

の512K×16SRAM領域を有する。好ましい実施形態で、イーサネットコントローラ360は、JTAGドライバとして機能し、JTAGポート341を通して連続的通信チャネルを提供し、イーサネット上に提供されるソフトウェアからのPM 10のFPGAを検定及び再プログラミングする。これにより、単にFLASHメモリバンク338に新しい構成ファイルをダウンロードすることによって後のシステムアップグレードが提供されることができるようになり、ビデオプロセッシング機能が、アプリケーションプログラム要件に基づいて変えられることができるようになる。特に、イーサネットコントローラ360はFPGAを含み、PM 10上にFPGAの再構成のためのPM 10上のJTAGコントロールを提供し、ブリッジ342-348、PCIブリッジ350、PCIアービタ352及びコントローラ及びPMCB 334（図示せず）のアービタ等の制御インタフェースを提供するようにプログラムされる。

#### 【0095】

PM 10が、VPSのビデオ処理をできる限り最適化するように設計されていたので、PM 10は、プログラムしているソフトウェア及びオーバーヘッドの最小量で、マルチタスキングマルチプロセッサ制御操作サポートするように設計されていた。本発明の実施形態に従うと、プログラムの実行、ローカルデータへのアクセス、ビデオ情報へのアクセスを含む、全てのマイクロプロセッサ集約型の操作は、マイクロプロセッサの各々に局所化される。これは、これらの強い操作のために利用されるグローバルVPSリソースの量最小にし、複数のマイクロプロセッサが所定のアプリケーションで使用されるときに性能を最大にする。上で言及したように、VSRAMは、マイクロプロセッサ12の各々に専用のビデオバスを提供することを通してこの原理の更なる延長を提供し、このように、ビデオハードウェアからマイクロプロセッサ12までメモリ及びバンド幅制約と関連するいかなるボトルネックも除去する。8-及び16ビットサインのVSRAMのメモリマッピング及びゼロ拡張読取り操作は、C40の32ビットのみの構成を有する変数幅データのVSRAMへのアクセスを最適化する。

#### 【0096】

PM 10上の共用リソースは、性能及び柔軟性を最大にするように慎重に考慮

された。例えば、PM 10の4つのUART 340は、これは、全てのマイクロプロセッサ12が、マイクロプロセッサからマイクロプロセッサへのオーバーヘッドなしで、外部デバイスと通じることを可能にするために、グローバルにアクセスできる。これも受け入れられる、それはUARTからのインタラプトのバンド幅及び周波数は、マイクロプロセッサ12のパワー及び関連PMバスの速度に比較して比較的小さいからである。

#### 【0097】

好ましい実施形態でPM 10も、リアルタイムビデオ処理用にハードウェア制御ライブラリ(HCL)を含む。更に詳細に下で説明するように、HCL構成により、VPSが、かなり多くのハードウェアを有する強力であり、かなり大きな、システムハードウェアの適度の量を有する比較的小さいシステムから滑らかにスケーリングされることができるようになる。加えて、デザインのモジュラ性は、VPSが、新しい処理コンポーネントの加算及び利用できる装置の再構成を通しての特殊アプリケーションに合うようにカスタム調整されることを可能にする。そのモジュラ及び伸長可能な構成を通して、そのHCLは、開発されることができVPSシステムの多様なセットに容易に適用できるようにする。上で言及したように、並列パイプライン化されたビデオ処理構成、例えば、VPSは、かなり速い速度を有する非常に最適化されたやり方でビデオ操作を実行することができる。例えば、VPS各ビデオストリームの33Mpixels/secの決定的速度で、像の複数のストリームを処理することができる。この性能は、特殊なハードウェアデバイスの使用を通して特にコンボリューション、単純なイメージポイントワイズ計算、ルックアップテーブル操作及び他のような操作を実行するように設計されているそのような操作を使用して他姓される。メモリーマップされたコントロール及びステータスレジスタを使用してビデオを処理するハードウェアデバイスは、一般的に制御される。異なる装置に対する制御レジスタへの書き込み値を通して、装置は、異なる操作を実行するように命じられる。

#### 【0098】

この例は、読込まれるフレームストア読み込み操作であり、イメージが処理及び／又はストレージのためのフレームストアから読込まれることになっている。



VPSを有するフレームストア読取り操作パラメーターの典型的なセットは、以下のものを含む。

【0099】

フレームストアから読込まれるイメージのサイズ。

【0100】

フレームストアメモリ内のイメージの位置。

【0101】

読み出される場合のこのイメージに対する水平ブランキング時間。

【0102】

読まれる場合のイメージに加えられることができる拡張（又はズーム）。あいにく、ビデオ処理操作の複雑なセットを実行するために、このビデオハードウェアを使用するアプリケーションを書込むプログラマを有するので、プログラムブルプログラマブル制御レジスタへの、レジスタ規格インラインメモリアクセスを使用することは、明らかに禁止である。この種類のプログラムすることから生じるコードは、全く読めない。フレームストアを使用するアプリケーションで、フレームストアデザインへのエラー又は変化は全ての位置に変更を必要とする。フレームストアレジスタの各セットの位置は、既知のアプリオリであり、それらがアプリケーションにおいて一旦固定されると修正されることができない。本発明の実施形態に従ってハードウェアデバイスをプログラムすることへのより合理的解決は、ハイレベルライブラリ、ビデオハードウェアをプログラムするPM 10のマイクロプロセッサ12の何れか又は全てにロードされるロードされるC-呼び出し可能機能を使用することである。プログラマは、各装置に対して一組の制御レジスタを備えて提示されるよりはむしろ、ハードウェアに指令するために要求される実際の操作を単純化又は「ハイド(hide)」するデータ構造、シンタックスを理解するために単に使用されるアプリケーションプログラミングインタフェース(API)が提供される。

【0103】

例として、フレームストアから読込み、他のフレームストアにそのイメージを保存する問題についてよく考える。手動でプログラムしている多数のコントロー

ルが登録するよりはむしろ、（この各フレームストア15のレジスタを上回るも  
とができる）、プログラマは、その代わりに、以下の一組の関数呼出しを使用す  
ることができる。

**【0104】**

```
par();
fsRead(image1);
fsStore(fsimgOutput(image1), image2);
endpar(fsimgVDEV(image2)).
```

**【0105】**

このシンタックスを使用して、何がこの制御ループに起こっているか理解する  
ことは、より明らかである。あるイメージ、（イメージ1と呼ぶ）は、ソースフ  
レームストア内に存在するように定義された。そのイメージについての情報（そ  
のサイズ、その位置、その他）の全ては、このイメージ構造でカプセル化される  
。転送先は、他のフレームストアで存在するイメージ2として定義される。par

（）関数は、ライブラリに、操作が並列に実行されるように指示する。操作の終  
わりのendpar（）関数は、3つの機能を果たす並列に実行されることになってい  
る一群の操作の終わりを合図すること、操作を可能にして、操作の実行を開始す  
ること、及び、その操作が完了するまで待つこと。endpar（）にパスされる引数  
は、イメージimage2のストアが完成するまで、機能が待つことであることを示  
す。

**【0106】**

fsRead（）及びfsStore（）関数は、フレームストアのプログラムしている制  
御レジスタの全てを扱う。コントロールレジスタをプログラムすることを要求さ  
れる情報は、image1及びimage2のイメージ定義にはめ込まれる。ソースから転  
送先までイメージを送り出すクロスポイントスイッチルーティングはfsimgOutput  
（）関数を通して取り扱われ、それは、第一のフレームストアのアウトプット  
が第二のフレームストアのインプットにルーティングされることになっているこ  
を示す。プログラマが、ビデオをルーティングするためにコントロールレジス  
タを明示的にプログラムすることから遠ざけ、及び、プログラマの操作でコンシ

ステンシーチェックができるようにする。

#### 【0107】

リアルタイムビデオ処理に有効なハードウェア制御ライブラリに適合しなくてはならない多数の厳しい要件がある。多分最も基本的にはHCLがリアルタイムフレームワーク内に動作されることになっていると認められなければならない、それは、ビデオハードウェア操作を実行するために要求される時間が、決定的且つ予測できなければならないことを意味する。同様に、ビデオハードウェア処理多重操作の複雑さのために、マルチタスキングの環境は、通常必要である。これは、プログラマがハードウェアの異なる部分を有するビデオ処理操作の異なる部分で同時に動作する並列タスクをセットアップすることをできるようにする。HCLの好ましい実施形態は、リアルタイムオペレーティングシステム(RTOS)の基金の上でつくられる。

#### 【0108】

加えて、ハードウェア制御ライブラリをプログラマに対して柔軟にしておくことに大きな注意が原割らなければならない。この柔軟性の例は、ソフトウェアのかなりの部分の容易な再使用、異なるビデオハードウェアタイプ及び数を有するシステムを取り扱うハードウェア制御ライブラリ有能力、それによって、アプリケーションに維持の過度に複雑及び困難性を引き起こすことがありうる複雑な低レベルシステム相互作用の効果的な隠蔽を含む。

#### 【0109】

ハードウェア制御ライブラリの最終の要件は、動作に効果的であるということである。マルチタスキングの環境内の、タスクスイッチングは、低いオーバヘッドで速くなければならない。コントロールレジスタのプログラム、するためにかかる時間は、コンシステンシーチェックの実行、及びハードウェア制御のためのソフトウェアでの他の管理作業タスクの実行は、オーバヘッドを低減するためにわずかに極小に保たれなければならない。そのマルチタスキングコントロール、インタラプト駆動操作(通信を含むこと)、セマフォ及びメールボックス、プリエンプティブ操作及び他のそのような特徴を提供するリアルタイムオペレーティングシステムの使用は、完全に効果的なHCLに供給されなければならない

基礎である。

#### 【0110】

幾つかのシステムは、(通常Cプログラミング言語と同様のシンタックスを有する) 翻訳された言語を使用し、ソースプログラムファイルの構文解析を通してハードウェア制御レジスタアクセスを発生させる。プログラムする方法は、ハードウェアのコントロールのプログラミングの「隠す」正味の効果を有し、そのうえ、最終の実行可能なシステムにコンパイルされるかなり効果的な、インラインコードを提供する。一般的に、ハードウェアアーキテクチャコンストレインツ及びHCLソフトウェアアーキテクチャの結合のため、これらのライブラリは、通常マルチタスキングの環境内のサポートされない。これは通常直接に引き起こされる。その理由は、提供されるコードが再インプット可能な操作—中断されるソフトウェア機能の能力として定義され、及び副作用なしで他のタスクによって再度呼び出される—をサポートしないからであり、又はソフトウェアが同時に操作を実行するように指示する複数タスクを取り扱うことができないハードウェア資源を取り扱うからである。コードを解析して及びコードを「再コンパイルする」不必要な他の副作用は、ハードウェアの動的制御が通常不可能であるということである。イメージサイズのようなパラメーター、実行されているハードウェア操作のシーケンス、及び、特殊アプリケーションによって使用される処理のタイプは、正しくコンパイルされるために前もって定義されなければならない。これは、イメージサイズ、関心の領域又はビデオデータに基づく測定又は決定に基づく事象のより動的なシーケンスさえ変える動的な処理を実行するHCLの能力を除去する。これは、リアルタイムビデオ処理アプリケーションを望ましい性能及び保全性で実行すること困難から不可能にする明らかな高速結果を有する。これらの限定は、通常ビデオハードウェアの下の構成、及び一定の遅延を取り扱うアプリアイミングのその信頼性、及びリアルタイムオペレーション中に効率的に決定されることができない資源管理による。

#### 【0111】

他のライブラリは、C及びC++のような高級言語から呼び出されることができ機能を提供する。再インプット可能な操作を取り扱うことができるが、これ

らのライブラリは通常そうするように設計されていない。更に、それらは、一般的にリアルタイムオペレーティングシステムのフレームワーク内に使用されるように設計されていないか、又はそれらがリアルタイムオペレーティングシステムへのリンクをサポートするならば、それらは、ハードウェア制御が、同時に実行しているタスクを使用して実行されることを許容しない。

#### 【0112】

ハードウェア制御ライブラリの異なるタイプでの特徴のこれらの欠如の正味の結果は、ビデオ処理ツールの強い及び効果的なセットを提供しないことである。マルチタスキングサポートの欠如は、プログラミングにあまり役に立たない連続した操作のシーケンスとしてコードが書込まなければならないことを意味し、同時に実行することになっている一定のタスクを、効果的に実行することがほとんど不可能にする可能性がある。実行されているイメージ処理操作のアプリオリ解析は、プログラマに前もって一組の処理コンストレインツを決定することを要求し、ビデオ処理ハードウェアのフルパワーが利用されることを許容しない。このように、本発明の実施形態のVPSのHCLは、これらの限界を克服するように設計されている。

#### 【0113】

当業者が、認識する本発明の実施形態のVPSハードウェアをプログラムする最も基本的な見地は、ストレージ装置、ビデオインプット装置及びビデオアウトプット装置間のビデオの移動である。搬送中に、ビデオフレームは、処理ユニットのいかなる数も通してパスされることが出来る通過するにつれてビデオデータを修正する。操作のこのタイプの結果は、イメージを処理するためにかかる時間は、搬送を実行するために要求される時間とほぼ同じである。ビデオのスループットは、常に同じままである。パイプラインの遅れは、彼らの操作を必要に応じて実行するために処理装置によって導入され、これらのパイプラインの遅れは、全体のイメージ転送時間と比較すると通常かなり小さい持続時間である。

#### 【0114】

本発明の実施形態のVPSで、ビデオは、クロスポイントスイッチ202の使用を通した異なるボードを通して流れる。ビデオが2つの装置の間で移動しなけ

ればならないたびに、クロスポイントスイッチ202は使用される。全てのビデオ装置インプット及びアウトプットは、このクロスポイントスイッチ202に接続している。そのようなハードウェアアーキテクチャの基本的な原則は、並列操作ブロックである。その並列操作ブロックのねらいは並列且つ同時に操作を開始するために一組の装置を構成することである。この一例は2つのイメージがフレームストアから読込まれ、共に加えられ、及び、結果の合計イメージは、第3の転送先フレームストアに保存されるプロセスであろう。プロセスユニット（この場合加算器）は、同時に（同期して）、2つのビデオ信号を受け取らず、2つの信号間のタイミングで差異を補償しなければならない。タイミング差異を低減するために、2つのフレームストアは、同期して読込まれる。前記の信号（RD\_\_START）は、同じクロックサイクル上でそれらの読込まれた操作を開始するように2つのフレームストアを指令する。これは、読取りの間に遅延がないことを確保する。このように、これらの2つの操作は、並列に実行されて、同じこと並列操作ブロック内で熟考される。一度始まると、ビデオ処理操作は更なる管理なしで完了する。

#### 【0115】

この操作を示す例は、図2に関して説明されることができる。例えば、フレームストア204及び206で定義される2つのイメージは、フレームストア読取りコマンドを使用して読込まれる。フレームストア204及び206のアウトプットは、次にCALU 228のイメージ加算器に、クロスポイントスイッチ202によって次にルーティングされ、それは、2つのインプットビデオフレームの上で、ポイントワイズイメージ加算を実行し、その結果アウトプットイメージを提供する。この合計イメージは、保存されるところで、転送先フレームストア（例えば208）に、クロスポイントスイッチ202によって次にルーティングされる。これを並列操作ブロック内で実行し、RD\_\_START信号で操作を始めることを通して、フレームストア204及び206から読込まれる2つのイメージは、同期して外へ読込まれることが保証され、このように、CALU 228のイメージ加算器のインプットでビデオタイミングの差異を導入しない。当業者は、2つのイメージが同時に開始することは、CALU 228に対して、重

要でないことを認識するであろう、その理由は、C A L U 228は自動的にタイミングでの差異補うからである。しかし、本発明の実施形態のV P Sの他の機能が、必ずしもタイミング差異（即ちP Y R 212-218及びP A L U 232）を、補償するというわけではない。

#### 【0116】

R D\_\_\_S T A R T信号が与えられると、フレームストアの異なるセット及び他の装置は、プログラムされ及び割込み可能にされることができる。第一の例の様に、R D\_\_\_S T A R T信号が提供されるまで、これらの装置は動作し始めない。これらの装置は、プログラムされ、割込可能にされることができ、一方、異なるハードウェアを有する他のビデオ操作がまだ実行され、他のR D\_\_\_S T A R Tのアサーションが、すでに進行中のビデオ操作を妨害しない。簡単な例は、ビデオ操作を実行するためにプログラマによって使用される基本的H C Lシンタックスを導入するために使用される。今説明したビデオ操作について考える。H C Lシンタックスにおいて、操作は以下の関数呼出しのシーケンスで実行されるであろう。

#### 【0117】

```
par();
fsRead(image1);
fsRead(image2);
aluAddImage(fsimgOutput(image1),fsimgOutput(image2),add_device)fsS
tore(vdevOutput(add_device)image3);
endpar(add_device)
```

このシーケンスは、次のように説明されることができる。

#### 【0118】

```
par();
```

この関数へのコールは、H C Lに、並列操作ブロックが始まることを指示する。

#### 【0119】

```
fsRead(image1);
```

fsRead(image2);

#### 【0120】

すでに、プログラマによってフレームストア204及び206で定義される2つのイメージは、彼らのフレームストアから読込まれるように指示される。フレームストア内のイメージの位置及びサイズを含むイメージの定義を使用すると、フレームストア読取り操作は、フレームストア装置204及び206で設定される。実際の読取り操作はまだ生じない。

#### 【0121】

aluAddImage(fsimgOutput(image1), fsimgOutput(image2), add\_device);

この関数は、多重操作を実行する。第一に、クロスポイントは、2つのフレームストア204及び206のアウトプットからCALU 228のイメージ加算器のインプットにルーティングされる。第二に、CALU 228のイメージ加算器は、適切にイメージを加えるようにプログラムされる。

#### 【0122】

fsStore(vdevOutput(add\_device), image3);

このコマンドは、2つの関数を実行する。第一に、それはクロスポイントスイッチ202を使用して、転送先フレームストア208のインプットに、CALU 228のイメージ加算器のアウトプットを接続する。第二に、転送先フレームがイメージのストレージに備えるために208を保存することは、プログラムする。

#### 【0123】

endpar(add\_device);

これは、並列操作ブロックを終了する関数である。この関数は、フレームストア204及び206から、読出しを開始するためにRD\_START信号を始動させ、加算器（すでに実行されたクロスポイントスイッチルーティングとして定義されるたように）を通して、次に自動的に送り出され、及びその後、転送先フレームストア208にストアされる。この関数は、RTOSがまた、タスク（これらの関数を呼び出しているタスクの更なる操作を終了する）を、加算器がその操作を完了するまでブロックすることを引き起こす。このように、他のタスクの



中の他の操作は、予定を変更されて及び加算操作が完成するまで処理を続けることができる。

#### 【0124】

本発明の実施形態に従って、ビデオ装置（VDEV）のコンセプトが、導入される低レベルで装置に関する下の情報を理解することなく、プログラマが装置を使用することを可能にするようにVPS内の異なる装置について情報をカプセル化する。例えば、前の例で使用されたadd\_device変数をとる。add\_device定義を使用して、プログラマはそのフル有効度に加算器を使用することができた。加算器の制御レジスタの位置、CALU 228のイメージ加算器のクロスポイントインプット及びアウトプット、及び加算器操作がいつ完成するかを決定するために使用されるインタラプト情報についての全情報が、add\_device定義でカプセル化され（HCL内のC構造として実行される）。

#### 【0125】

そのVDEVは、C-構造としてHCL内で好ましくは実行される。VPS内の全てのビデオハードウェア構成機器、フレームストア、ピラミッドモジュール、CALUs、PALUs、ドータボード装置、その他を含むことは、1つの（又はそれ以上の）VDEV構造を有する、HCL内で全て表される。好ましくは、イメージが存在するフレームストアに対して、フレームストアイメージ定義は、定義内にVDEVを含む。このように、イメージと一致してVDEVにアクセスすることを通して、フレームストアについての低レベルの情報が決定されることができる。異なる装置は、制御レジスタ、クロスポイントインプット及びアウトプットの異なる構成、及び他の独特な特徴を有する。幾つかの、全てのVDEV構造の部分は、全ての装置一般の特徴である（その最も基本的な部分は、どちらがVDEVで表されるビデオデバイスタイプを示すフィールドである）。

#### 【0126】

他の特徴、例えば制御レジスタ又は他の装置—独立メモリ領域の位置は、VDEV構造の装置—特殊フィールド内で保存される。異なるHCL関数には、異なる装置が提供される。例えば、イメージ加算器を操作する関数は、ビデオ、読取り及び記憶操作を実行するためにフレームストアを操作するものと異なる。その

V D E Vは、装置に操作されるのを可能にする指針及び他情報を提供する、及び装置を動作するために提供される関数は、装置を指令するためにV D E V情報を使用するように使用される方法を含む。フレームストアの場合、V D E Vは、フレームストア制御レジスタが、フレームストアコントローラ240内のどこに位置するかを指示するが、フレームストア以外の、関数（例えばfsRead（）及びfsStore（））は、ビデオ読取り及びビデオ記憶を実行するためにフレームストアレジスタを操作するノウハウを含む。

#### 【0127】

一般の構造にビデオ装置をカプセル化するこのアイデア及び、V D E V sを操作するための装置関数を使用することは、H C Lに重要なコンセプトである。この構成は、V P S ドータボードとして通常実行され、他の装置を操作するために使用される方法を変化することなくH C Lに加えられる新しい装置を可能にする。一般のイメージフォーマット及び一般のルーティングリソース（クロスポイントスイッチ202）を使用して、これらの装置へのビデオデータインタフェースが定義され、装置を動作する全ての追加のノウハウは、装置の上で動作する新しい関数に配置される。このまた、システム内の存在する装置に従ってスケーリングするH C Lを使用可能にする。例えば、与えられたV P S システムが、装置の一定のタイプを有しない場合、プログラマは、その装置を操作する関数を使用しようとしな。順番に、この、これらの関数がプログラマの最終のアプリケーションにリンクされないことを意味する。このように、プログラマのアプリケーションは、これらの使用を実行することを要求される関数を含むのみであり、また、価値あるメモリ空間を利用する他の余分な関数を含まない。

#### 【0128】

V P SがそのP M 10の上でちょうど1つのプロセッサを含まないので、H C Lは、効果的なこれらの複数のプロセッサのうちの1つ以上で、同時に実行されることができなければならない。そのようなマルチタスキング構成内で効果的にH C Lランを有するために、H C L関数は、再インプット可能でなければならない。そのH C Lは、これを考慮して設計され、全ての関数は、再入可能であるか、互いに排除され、再び入れられる可能性から適切に守られる。これを以下に

説明する。VDEV構造及び異なるビデオ装置の上で動作する関数を有するその使用は、各タスクが、ビデオ操作についての全関連情報をパスすることを可能にして、静的に割り当てられたデータを使用するよりはむしろタスクスタックフレーム上のHCL内で機能する。これは順番に、HCLが容易に再出可能に作られる。更に、VDEV構造は、操作中のいかなる方法にもおいて修正されない。このように、同じプロセッサの複数タスクは、困難性なしで同じVDEVsを使用することができる。

#### 【0129】

HCL内のマルチタスキングを有する1つの可能性の問題は、並列操作ブロック内のRD\_\_START信号の使用である。ハードウェアのプログラミング及びRD\_\_STARTの使用は、独立して動作しているタスクの間で互いに排除されなければならない。この排除が実行されないならば、1つのタスクによるRD\_\_STARTのアサーションは、これもハードウェアをセッティングする他のタスクによって設定される操作を早まって開始するかもしれない。この相互排除は、ソフトウェアの大部分のリアルタイムオペレーティングシステムによってサポートされることができ、セマフォ、メールボックス又はそのようなものが他の構成するを使用して、しかしこれはかなりのオーバヘッドを負う可能性がある。マルチプロセッサが単一の排除セマフォをパスする必要があるとき、このオーバヘッドは悪化する。この問題を軽減するために、ハードウェアセマフォは、PM 10内に提供される。上で言及されるのでのように、このハードウェアセマフォは、全てのマイクロプロセッサ12に対して可視であるPM 10のリード/ライトレジスタ343である。セマフォの上の所定の動作は、言及された。

#### 【0130】

このセマフォは、複数タスク及び複数のマイクロプロセッサ12の相互排除を以下の方法によって使用可能にする：

1. セマフォを必要としているタスクが、一意の値をセマフォレジスタ343へ書込む。
2. タスクが、セマフォレジスタ343から値をリードバックする場合、タスクは、セマフォを与えられる。タスクは、次に望ましい操作を実行して及び次に

ステップ4へ行く。

3. タスクが読込むことができないならば、セマフォからその値を後退させない場合、タスクは、ブロックし（産する）、他のマイクロプロセッサ12からインタラプトを待つ。このインタラプトは、セマフォが開放されたというメッセージのみを送り出すために使用されるだけである。セマフォが開放されたインタラプト信号がマイクロプロセッサ12で受け取られるとき、マイクロプロセッサ12は、ステップ1に戻る。

4. 12が有するマイクロプロセッサがその望ましいタスクを完了したとき、0の値を、セマフォレジスタ343に書き、全てのマイクロプロセッサを中断する。そして、その処理は続く。このようにこのセマフォを使用することによって、全てのタスク及び全てのマイクロプロセッサ12がハードウェア制御レジスタをプログラムする能力を自由に決られるようになり、RD\_START信号を使用する。par () 関数は、シングルプロセッサ上のビデオハードウェアへのアクセスを直列化する。マルチプロセッサは、装置が使用可能にされず、又はGVBポートが必要でない限りビデオハードウェアを同時にプログラムするだろう。プロセッサが装置を使用可能にし、又はGVBポートを得なければならない場合、セマフォを得ることによってこれらの同じ操作を実行することから他のタスクを排除しなければならない。endpar () 関数が呼び出されるとき、RD\_START信号は、主張される、セマフォは、開放され、ビデオ処理操作が完成していることを示してインタラプトが受け取られるまでタスクはブロックする（産する）。特に、endpar () 関数が呼び出されるとき、指定されたハードウェア装置（endpar () 関数をパスするVDEV引数によって指示された）がインタラプトを発生させるまでRTOSは、ブロックするタスクを引き起こす。ビデオ装置によるインタラプトは、それらの完了で発生させられるのみであるので、不必要なインタラプトは発生させられない。endpar () を呼び出すタスクがハードウェア操作の完了を待つ一方、タスクのこのブロッキングは他のタスクが処理を続けることができるようにする。

#### 【0131】

特定の操作の結論を決定するためにハードウェアデバイスのステータスレジス

タをポーリングすることは可能だが、ポーリングすることは、マイクロプロセッサのリソースの効率的利用を提供せずまたマルチタスキングのフレームワーク内で適切でない。このように、ポーリングすることを使用するよりはむしろ、本発明のビデオ装置は、マイクロプロセッサ12の各々に次に多重化されるインタラプトを発生させる。発明の好ましい実施形態で、上記に言及したのでのように、最高6つのVPM 20を有することができるVPSの各VPM 20によって発生されうる14の異なるインタラプトがある。また、上で言及したように、インタラプトフラグ及びマスクレジスタ244は、各VPM 20の上で利用でき、インタラプトを処理して及びマスクされていないインタラプトをPM 10にパスさせる。PM 10は、故に各VPM 20ににただ1つのインタラプトだけを受け取る。

#### 【0132】

明らかに、タスクは、待たれている装置を含む、所定のVPM 20からのインタラプトを待っているとき、タスクは、VPM 20からのいかなるインタラプトもタスクが待っている装置を示すと仮定することができない。VPM 20のインタラプトを処理する割込みサービスルーチンは、このように、VPM 20からの所定のインタラプトが望ましいインタラプトであるか否かを決定するためにVPM 20のフラグレジスタ244を検査する。望ましいインタラプトが発生させられたならば、マスクレジスタ244はオフに（インタラプトが再び偶然に発生させられないように）され、フラグレジスタ244はクリアされる。

#### 【0133】

VPM 20の上で割込みマスク及びフラグレジスタ244を同時に操作するタスクの間で生じる競合状態の機会があるので、標準読み込み変更記憶操作を通すよりもむしろ、単一のインタラプトと一致している個々のマスク及びフラグビットがセットされることができ、原子的にクリアされることができるよう、マスク及びフラグレジスタ244は設計される。VPM 20のインタラプトレジスタ244のこの原子操作は、一度にただ1つのタスクが、所定のVPM 20の上で所定の装置を待つ仮説に結合し、VPMインタラプトレジスタ244で生じることができる全ての潜在的競合状態を除去する。

## 【0134】

VDEVが所定の装置の状態（どのルックアップテーブルバンクがすでにプログラムできるLUTに割り当てられたかなどの）について、情報を含むまれな場合には、状態情報はPM 10の共有DRAM領域336で格納されて及びポインタによってVDEVにリンクされる。DRAM領域336へのアクセスは、互いに排除されるpar () 及びendpar () 関数で使用するハードウェアセマフォに同様であるハードウェアセマフォを使用すること。この性質のアクセスは一般的にアプリケーション実行の時間—臨界ループ内でなく、最初のスタートアップ及びアプリケーション中だけのノーマル環境の下で生じ、このセマフォは、タスクによってポーリングされる。相互排除を必要としていない他のDRAM操作は、セマフォを利用しない。

## 【0135】

本発明の実施形態のHCLは、マルチプロセッサをサポートする予定である。しかし、ハードウェアが処理のために使用されうるまえにハードウェアを検出及び初期化するために実行される必要がある多数の操作があることが認められる必要がある。また、PM 10の全てのマイクロプロセッサ12のスタートアップは、特にパワーオン初期化でにおいて、競合状態が生じないことを確保するために慎重に編成される必要がある。「原始の」スタートアップは、HCLが実行するスタートアップの第一のレベルである。原始のスタートアップ中に、1台のみのマイクロプロセッサ（初期値マイクロプロセッサ302）だけは、VPSハードウェアを初期化するためにステップのシーケンスを実行する。原始のスタートアップのステップ及びこれらのステップが重要である理由を、今説明する。

## 【0136】

マザーボード及びドータボードの数及びタイプを変えることがありうるように本発明の実施形態のVPSが設計されるので、HCLは、存在する構成を知っていなければならない、利用できるハードウェアを自動的に検出することによって存在する装置の全ての可能な置換を収容することができなければならない。好ましくは、全てのVPM 20及びVPDは、特殊制御レジスタにアクセスすることを通して唯一に識別されることができる。マザーボードスロット又はドータボー

ドサイトでのボードの存在又は不在は、また、識別されることができる。原始のスタートアップルーチンは、VPM 20の全ての可能なスロットに問い合わせる、それから、ドータボードを見つけるために各検出されたVPM 20上の全てのドータボードサイトに問い合わせる。

#### 【0137】

ビデオ装置が検出されるとき、原始のスタートアップは、DRAM 336でVDEVを割り当て、VDEVに検出された装置適切な装置情報を配置する。このように、原始のスタートアップの終わりに、VPSで検出された装置の全てを説明するVDEVsの完全セットは、グローバルDRAM 336に存在する。この情報がグローバルDRAM 336であるので、全ての装置はその情報にアクセス可能でそれを使用することができる。

#### 【0138】

大部分のビデオ装置は、ある種のパワーオン初期化を必要とする。装置が検出され、VDEVsがグローバルDRAM 336装置に割り当てられるとき、処理が開始する前に、それらが既知の及び有効な状態においてあることを確保するために装置は原始のスタートアップによって初期化される。

#### 【0139】

PM 10上に存在する他のマイクロプロセッサ12は、原始のスタートアップ中のビデオ装置にアクセスすることを防ぐ。マイクロプロセッサ302が原始のスタートアップを完了するとき、他のマイクロプロセッサに適切な信号を送る。一度生じると、全てのマイクロプロセッサ12は、彼らのスタートアップを開始する。このスタートアップは、HCLの実行に必要なマイクロプロセッサ12に局在する全ての情報を例示化する。好ましくは、各マイクロプロセッサ12は、原始のスタートアップ中に確立されるVDEV情報の全てのそれ自身のローカルコピーを保ち、及び、全てのVDEV情報は、そのローカルデータSRAM領域に各マイクロプロセッサ12でコピーされる。

#### 【0140】

VDEV情報を各マイクロプロセッサ12に局在させておくことは、HCLの効果的な操作決定的である。全てのマイクロプロセッサ12がHCLの実行可能

なコピー及びマイクロプロセッサ12への装置情報局所の全てを有するので、プログラムしているハードウェア及び他の操作は、マイクロプロセッサのローカルバスの外でアクセスの最小の量を有する実行されることができる。これは、PM 10及びVPM 20の一般のバスを別な方法では必要だったデータトランザクショントラフィックの多くから離しておく。このストラテジーを使用すること、實際上、PM 10及びVPM 20グローバルバスのトラフィックの全ては、ただハードウェアレジスタプログラミングのためである。他のトラフィックだけは、ビデオ装置についてのグローバルデータDRAMに臨時のアクセスを含む、及び外部のコンピュータと通信するためにマイクロプロセッサ12を使用可能にする通信リソースにアクセスする。

#### 【0141】

VDEV情報が各マイクロプロセッサ12のローカルSRAMにコピーされた後、マイクロプロセッサ12は次に彼らのタスクを始める。タスクの各々は次にビデオ操作を実行するためにローカルSRAMで存在するVDEV情報を使用することができ、実行は開始される。各タスクは、一組のVPSビデオ装置リソースを使用するために定義される。いかなるタスクでもいかなるビデオ装置にもアクセスすることができるので、所定の時間でのビデオハードウェアのどの部分にどのタスクがアクセスすることができるかの決定がプログラマに残される。HCL内の及びVPSハードウェアの全ての実行が実行時で決定的であるので、プログラマは、正確に異なるタスクが異なる装置を使用する時間をセグメント化することができる。リアルタイムオペレーティングシステム環境によって提供される多重タスク及びマルチプロセッサ制御セマフォは、プログラマが、均一な態様で他のタスクでタスク実行を調整することを可能にする。あいにく、これは、アプリケーションプログラマの責任である。それは、実行時アロケーションに対して要求される管理作業の量によるからであり、多数の装置の検査は、HCLがリアルタイムで実行することができるようにするにはあまりに高かった。

#### 【0142】

しかし、本明細書で説明されるクロスポイントステートマシン238は、この管理作業要件を軽減する。各VPM 20の上で存在するクロスポイントスイッ



チ202は、ビデオ操作と共にビデオ装置を接続することに対して最初に責任がある。いかなる1つの時間もリソースのオーバラップしていないセットを使用する異なるタスクに、プログラマが異なるビデオ操作をセグメント化し、VPM 20のクロスポイントスイッチ202が装置間の接続のいかなるセットも取り扱うのに十分大きいから、単一のVPM 20上の装置間にクロスポイントルーティングリソースを動的に検査及び割り当てる必要性はない。しかしVPM間で通じているために存在するクロスポイントチャンネルは、グローバルビデオバックプレーンを使用して、ダイナミックアロケーションを必要とする。HCLは、HCLの内部のVPM 20間にトランザクションを「隠す」し、プログラマは、明示的にアクティブなバックプレーンルーティングリソースを割り当てる必要がない。しかし、バックプレーンは、バンド幅の有限の量を有するルーティングリソースであり、従って、バックプレーンビデオチャンネルの競合がありえる。

#### 【0143】

各VPM 20でのクロスポイントステートマシン238は、バックプレーンルーティングリソースを動的に割り当てるメカニズムを提供する。クロスポイントスイッチステートマシン238は、クロスポイントチャンネルがいつ使用中で又は現在利用できるかについて示す。相互排除を通して、所定のポイントでの適時にビデオハードウェアは、全てのマイクロプロセッサ12上でただ1つのタスクだけがプログラムしているのでクロスポイントステートマシン238に競合状態がない。このように、VPSボード間でビデオをルーティングするために所定のタスクが必要の場合、ソースVPSボードからの利用できるアウトプットの各々は、クロスポイントステートマシン238を通して問い合わせされる。全てのアウトプットが現在利用されるならば、HCLは中断する。別な方法で、ある利用可能資源は、ルーティングのために使用され、このリソースは、ビデオトランザクションが完成するまで、ステートマシン238によって使用中として自動的にマークされる。

#### 【0144】

その結果、クロスポイントステートマシン238の使用は、ボードからボードへのビデオ伝達の取扱いに対して優れている。このステートマシン238なしで

、HCLはグローバルにアクセスできる位置で等価情報を維持しなければならないであろう。このビデオ搬送がされた場合リソースを開放することを必要とするであろう。これを行うことは、追加の管理作業必要とし及び、endpar () で必要とされるプロセッサ操作の数を増加させるであろう。

#### 【0145】

##### ビデオ処理データボード

ビデオ処理データボード (VPD) は、専門化された関数を提供するために各VPM 20に加えられる。各VPDは、制御バスに加えて、直接にVPMクロスポイントスイッチ202に接続している多数のビデオインプット及びアウトプットポートを有する。VPM 20の現在好ましい実施形態で、2つのVPDが、取り付けられることができ、各VPDは最高6つのビデオインプットポート及び6つのビデオアウトプットポートを有する。VPM 20の現在好ましい実施形態で実行される4つのサンプルVPDを、今、図4-8に関して説明する。

#### 【0146】

図4は、図1でブロックフォームで示したディスプレイVPD 22を示す。表示VPD 22は、市販ビデオエンコーダ402 (フィリップスSAA7182) に基づき、それは、コンポーネントデジタルビデオ信号を、標準コンポジットビデオ信号 (CVSB)、コンポーネントビデオ信号 (S-ビデオ) 及びRGBに変える。VPM 20のフレームストアが一般的にアプリケーションプログラムでハイデマンド (high demand) であるので、ディスプレイVPD 22は、含まれる3フレームストア404-408を有する。フレームストア404及び406は、各々UV (Chroma) イメージ及び、コンポーネントデジタルビデオ信号Y (輝度) イメージを緩衝するために使用される。4:2:2標準にサンプリング定義に従うと、UVフレームストア404は、2つのカラーイメージU及びVを含み、そしてそれは、水平インターリーブされた態様でストアされている。フレームストア408は、ビデオディスプレイの情報のグラフィックスオーバーレイを提供する。好ましくは、フレームストア404-408は、全て1k×1.5×8ビットサイズである。従って、フレームストア404-408は、ディスプレイイメージのダブルバッファリングを提供するように十分大きい。ダブルバッファ

リングでは、新しいイメージデータは、フレームストアの1つの部分にロードされることができ、他の部分はディスプレイの外に読み出される。ディスプレイへのイメージレーテンシが関心である場合、ディスプレイバッファへのイメージデータの搬送は、時間と共に時間を計られることができるディスプレイイメージアーティファクトなしで、イメージデータは同じことバッファ領域から読み出される。

#### 【0147】

2つのフィールドプログラマブルゲートアレイ (FPGA) は、ディスプレイ VPD 22 上で提供される。第一のものは VPD コントローラ 410 であり、第二のものはビデオコントローラ 412 である。VPD コントローラ 410 は、FIFO 414 を通してマイクロプロセッサ制御バスにアクセスを提供することによって VPM 20 に制御インタフェースを提供するそのうえ、単純なグラフィックス関数 (例えばイメージクリア、線画及び活字生成) のグラフィックスコントローラ 416 を含む。ディスプレイ VPD 22 上のこれらの関数を有することは、GCB 40 上のデータバンド幅のかかなりの低減を提供する。他方、ビデオコントローラ 412 は3つのフレームストア 404-408 のアウトプットとビデオエンコーダ 402 との間にあり、図形変換表を制御すること。それは、また並列 D1 アウトプットインタフェース 418 又は他のどのデジタルアウトプットインタフェースの制御も提供し、フレームストア 408 から輝度イメージに適切にオーバーレイイメージを加える加算器 420 含む。

#### 【0148】

図5は、図1においてブロック図で示したデジタイザ VPD 24 を示す。デジタイザ VPD 24 は、フィリップスチップ SAA7111A ビデオデコーダ 502 に基づき、複合 (CVBS) 及びコンポーネントビデオ (S-ビデオ) データ (NTSC 及び PAL 両方とも) のカラー解読及びデジタル化提供する。3つのチャンネルは、デジタイザ VPD 24 上で好ましくは実行され、3つの非同期カラービデオチャンネルをサポートする。デジタイザ VPD 24 も、別個のビデオデコーダ 502 で、各カラーコンポーネントをデジタル化することによって RGB インプットをサポートする。並列 D1 又は他のデジタルインタフェース 504

は、また、並列D1又は他のデジタルインプットを取り扱うために含まれる。

#### 【0149】

ビデオデコーダ502は、60Hz（NTSC）での720×242のフィールド又は50Hz（PAL）での720×288フィールドでデータをデジタル化する。2台のデジタルチャネルはアウトプットに送られ、1つは輝度（Y）のみ、1つはインタリーブされたU及びVカラーコンポーネントである。これは12.5MHz SMPTE及び同様の基準によって定義されるように4：2：2のフォーマットのデータを提供する。各ビデオデコーダ502は、2つのフィールドバッファによって続けられ、それは輝度チャネルに対してはYフィールドバッファ506、カラー（U、V）チャネルに対してはUVフィールドバッファ508である。バッファ506及び508は、ビデオデコーダ502から13.5MHzデータをバッファリングすることによってVPM 20に最適化された処理を提供し、VPMフレームストア及び／又は他の処理要素に、VPMシステムクロック（33MHz）でのフルフィールドを読込むことが続く。

#### 【0150】

加えて、データが、VPMクロックスピードでフィールドバッファ506及び508から読込まれることができる場合、13.5MHzでフィールドデータのビデオ書込みに「追いつく」ことなく、システムコントローラに指示するプログラムできるインタラプトが、好ましくは提供される。これは、デジタイザVPD 24からビデオデータの呼び出し時間を最小にする一方VPM 20処理関数及びデータ極大スループット提供する。

#### 【0151】

図6は、図1においてブロック図で示した相関器VPD 26を示す。相関器VPD 26は、速い動き推定、スタビリゼーション及びイメージ融合に対して設計される。それは、VPM 20のSRAM及びFIFO 230を有するCALU 228と同一のFIFO 230及びSRAMを有する3つのCALUs 228を含む。加えて、CALUs 228のアウトプットは、各々PYR-2つのピラミッド処理ASIC 602及びLUT 604によって続けられ、VPM 20のPYR-2及びLUT結合（例えば212、220）と同様であ

る。動き推定及び融合の多くのアプリケーションは相関又はCALU 228で実行される計算に続くフィルタ操作を必要とする。2つのビデオデータが流すパスルーに、PYR-2つのASIC 602は、また、セットされることができる。

### 【0152】

図7は、図1においてブロック図で示したワーパVPD 28を示す。ワーパVPD 28は、リアルタイムパラメトリックイメージワーピングのために設計される及び、同時のイメージ取得及びワーピングのために2つのSRAMバンク702、704を含む。ワーパVPD 28は、アドレスジェネレータ706を使用しているパラメトリックイメージ変換のために、アドレス生成も実行する。好ましい実施形態で、変換は、アフィン（6つのパラメーター）変換で、アドレスジェネレータ706は、2つのFPGA（Altera EPF10K70）として実行される。これらのFPGAは、バイキュービック（bi-cubic）変換又は投影の変換（2つのアフィンの分割）の実行をサポートすることができるように十分大きい。任意の32ビットフローフィールドインプット（Xに対しては16ビット、Yに対しては16ビット）が、パラメトリック変換に加えられることができるVPM 20（即ち、4つのフレームストアから）に対して、4つのビデオデータストリームを受け取ることによる加算器708によって。発生させられたフローフィールドは、また、4つのビデオデータストリームとしてVPM 20に送り出されることができる。好ましくは、ワーパVPD 28は、1/32ピクセル解像度に正確である二本の線の補間回路710を含む。

### 【0153】

図8で示すように、ワーパVPD 28の現在好ましい具現化は、2つのSRAMバンク702、704及び3つのFPGA 802-806及び合計5つのビデオインプット及び5つのビデオアウトプット、から成り、本質的にかなり強力な可変関数モジュールにしており、標準操作はイメージワーピングのためにプログラムされているが操作の多くの他のタイプはプログラムされることができる。FPGA 802（好ましくはBGA356パッケージの中のEPF10K5

0)は、2つのSRAMバンク702、704の及び1つのビデオインプット及び1つのビデオアウトプットへの全てのインタフェースを含む。ワーパ関数に対して、FPGA 802はイメージを書込んで及びSRAMからイメージを読み込むためのSRAMに全ての機能的なインタフェースを実行する。FPGA 802は、クロックサイクル毎に4つの隣接したピクセル読み書きすることができる。4つの隣接ピクセルを読み込むことは、イメージワーブ操作の一部として二本の線の補間をサポートする。4つのピクセルの書込みは、また、ビデオデータ（高品質のイメージワーピングに必要としているように）のアップサンプリングをサポートするであろう。1つのSRAMバンクを32ビット「固定された」フローフィールドとして構成することも可能であり、一方、ワーブ関数のイメージ検索及びストレージのために他のSRAMバンク及びイメージの検索を使用する。この場合、他の2つのFPGA 804、806は、必要ではなく及び、除去されることができ又は他の操作に使用されることができ。FPGA 804及び806は、各々好ましくはEPF10K70（RQFP240パッケージで）である。ワーピング関数に対して、これらのFPGA 804、806のうちの1つは、Xアドレス発生及び他はYアドレス発生を実行するために専用とされる。スケールダウンされたアプリケーションに対して、X及びYアドレス生成は、1つのFPGAで実行されることができ他のFPGAは除去されることができ。より要求する関数に対して、全てのFPGA 802—806は、新しいワーパVPD 28を設計することなく、Altera（例えば使用した同じパッケージ内の10K100）から大きいFPGAと取り替えられることができる。これに対して、ワーパVPD 28もジャンパー線を変えることによって、3.3V操作に変えられることができる。

#### 【0154】

VPM 20の他のFPGAと同様に、ワーパVPD 28の全ての3つのFPGA 802—806は、JTAGチャネルを通したソフトウェアから構成されることができ。

#### 【0155】

グローバルビデオバスGVB

30を図1で示し、VPM 20及びPM 10への接続を図2及び3にそれぞれ示す。GVB 30は、VPM 20の上でクロスポイントスイッチ202の間でビデオを提供する方法を提供し、VPM 20のうちの1つからPM 10までビデオを提供することができる。GVB 30のビデオフォーマットは、ビデオフォーマットがVPSの残りを全部使用したのと同じである。

#### 【0156】

本発明の好ましい実施形態で、VPSは2つのバックプレーンを含む：1つはCompactPCI<sup>TM</sup>用であり、Global Control Bus (GCB 40)として役立ち、他はGVB 30用である。そのGVB 30は、いかなる一般のバス内を流れている基準に一致させる必要はなく、異なる幾つかの具現化として提供することができる。GVB 30の最も単純なバージョンは、ボードの間のポイントツーポイントハード接続であり、ルーティングされたプリント回路基板(PCB)バックプレーンとして具現化されたものである。このGVB 30は、「受動的な」GVBとして知られ、ボードの間で相互接続の固定したトポロジーを有し、VPM 20の予めデザインされた数をサポートするのである。VPM 20からGVB 30へのビデオのルーティングは、VPM 20の各々の上でクロスポイントステートマシン238をモニタして、実行される。実行中の、PM 10のハードウェア制御ライブラリは、モニタすることができるアウトプット及びインプットバスのどれVPMからのGVB 30へ／から、20は現在使用されている。このように、クロスポイントステートマシン238を使用して、VPSボードの間のビデオ信号のオンデマンド動的ルーティングは、達成されることができる。これは、使用され且つ利用できるGVB接続のソフトウェア管理作業に依存するその他方法に対する、GVB 30へのビデオルーティングの最も効果的な方法であり、順番にマイクロプロセッサ及びタスクの全ての中で与えられることが必要なグローバルデータのかなりの量の取扱い及び更新処理に依存する。

#### 【0157】

好ましい実施形態では、受動的なGVB 30の2つのバージョンがある：1つは2のVPMと1のPMで、1つは3のVPM及び1のPMである。しかし、

他の上の1つの受動的なバックプレーンの使用は、特殊用途であり、所定のタスクを実行することが要求されるVPMの数に依存する。3つ以上のVPM 20が単一のVPS内の使用されるとき、受動的なバックプレーンは、大部分のビデオアプリケーションにもはや十分でない。VPSでのボードからボードのビデオ移送の量は、実行されるタスクにかなり特殊であり、及び、各アプリケーションに対して異なる受動バックプレーンを有することは扱いやすくない。複数の受動バックプレーンの代わりに、アクティブバックプレーンが、ボードからボードへのビデオルーティングの問題を改善するために使用されることができる。

#### 【0158】

アクティブなGVB 30は、二次のクロスポイントスイッチ（この場合56×56×10ビットクロスポイントスイッチ）を使用し、バックプレーンの裏に取り付けられたドータボードとして存在する。このクロスポイントスイッチはGCB 40にブリッジされる制御レジスタを通してプログラム可能であり、単一ルーティング装置を使用している全てのVPSボードから、ビデオインプット及びビデオアウトプットの全てを接続する。アクティブなGVB 30の使用を通して、クロスポイントスイッチングメカニズムの利点の全て：ファンアウト、ノンブロッキング操作、その他は、非常に効果的に使用されることができる。トポロジーがもはやフィックスされないので、アクティブなバックプレーンは、異なるアプリケーション及びタスクの最も広く可能な種類をサポートする。アクティブなGVB 30の唯一の制限は、GVB 30へのVPSボードの各々への、そして、そこからの、ビデオパスの数である。好ましい実施形態で、GVB 30はチャンネル毎に33のMByte/secでのビデオ最高56チャンネルを収容することができる。

#### 【0159】

##### グローバル制御バス

グローバル制御バス最終のVPSコンポーネントは、GCB 40である。GCB 40の目的は、ランダムアクセスデータバスを提供することであり、それは、PM 10上のマイクロプロセッサ（C40 DSP）12を、VPM 20上に及びアクティブなGVB 30上に存在する制御レジスタにアクセスする



ことができるようにシステム内に存在しなければならない。G C B 40に関係するキーは、制御レジスタストア又はリード操作を実行する時間である。V P Sのリアルタイム制御オーバーヘッドを最小に保つために、P Mのマイクロプロセッサ12での制御レジスタへのアクセスは、できるだけ速くなければならない。P M 10の制御レジスタの多くの解析は、V M Eバスのような周知であるが遅いバスが、V P Sに必要な速度で、リアルタイムタスクを取り扱う十分な性能を提供しないことを迅速に示す。これらの関係する性能のために、Compact P C I<sup>7</sup>バスは、本発明の実施形態のG C B 40として実行された。Compact P C I<sup>18</sup>は、33MHzクロックを使用して動作し、250のnsecの必要な速度での実行しているランダムアクセスデータトランザクション/ストア操作ができ、それはV P Sの性能要件である。(レジスタ読み込みアクセスは、書込み操作よりも、リアルタイム制御中では一般でなく、全体的なシステムパフォーマンスに対して重要でない関係である。)

幾つかのバスがV P S内のあることを心にとめておくこと、マイクロプロセッサが制御レジスタにストア操作を完了するためにかかる、時間に関する重要な問題がまだある。制御レジスタへの全てのマイクロプロセッサアクセスは、P M C B 334の上に任意でなければならない、それから、Compact P C I<sup>18</sup> G C B 40の上に送り出され、それから、そのボードのローカル制御バスを通して転送先ボードの上で制御レジスタに伝搬され、次に実際に、宛先レジスタで格納される。この問題を改良するために、ポスト-書込みが、書込みプロセスの多数のステージで実行され、レジスタストア速度に最小の影響で、長く続く一連の連続レジスタの書込みを実行するためにマイクロプロセッサを解放する。

#### 【0160】

本明細書で教示される操作の方法及び装置が、本発明の実施形態の一例であることは、理解されるであろう。変更は、本発明の精神又は範囲から逸脱することなく、当業者によって容易に工夫されるだろう。ハードウェアで実行されるように、部材の多くは説明されたが、例えば、もちろん、既知の技術を使用してソフトウェアでエミュレートされることもできる。また、システムのパワーアップの後、ボード上のE P R O Mから特殊なハードウェア関数用に自動的に構成される

フィールドプログラマブルゲートアレイ（FPGA）を使用して、本明細書で説明したビデオ処理システムのその多くの装置関数及びコントロールインタフェースが、実行されるだろうことは、当業者によって認識されるであろう。大部分のそのようなFPGAは、また、JTAGチェーンに接続しているだろうFPGAの各々は、PM 10のマイクロプロセッサ12のソフトウェアからの異なるか又は修正されたハードウェア関数のために変更されることができる。全てのそのような変更は、添付の請求項の範囲内で含まれることを意図している。

【図面の簡単な説明】

【図1】

図1は、本発明の実施形態のモジュラのビデオ処理システムのブロック線図である。

【図2A】

図2Aは、アウトポート（XB\_\_\_\_OUT）に対するクロスポイントスイッチのクロスポイントステートマシンの状態をしめした図である。

【図2B】

2Bは、インポートポート（XB\_\_\_\_IN）に対するクロスポイントスイッチのクロスポイントステートマシンの状態をしめした図である。

【図2】

図2は、ビデオ処理マザーボード（VPM）のトップレベルのブロック線図である。

【図3】

図3は、プロセッサマザーボード（PM）のブロック線図である。

【図4】

図4は、ディスプレイビデオプロセッサデータボード（VPD）のブロック線図である。

【図5】

図5は、デジタイザビデオプロセッサデータボード（VPD）のブロック線図である。

【図6】

図6は、相関器ビデオプロセッサデータボード（VPD）のブロック線図である。

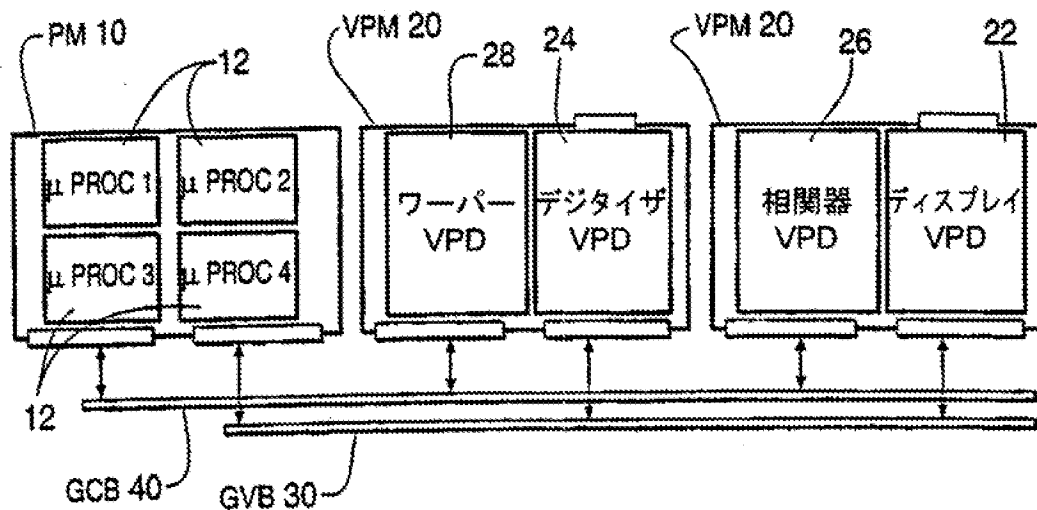
【図7】

図7は、ワーパビデオプロセッサデータボード（VPD）のブロック線図である。

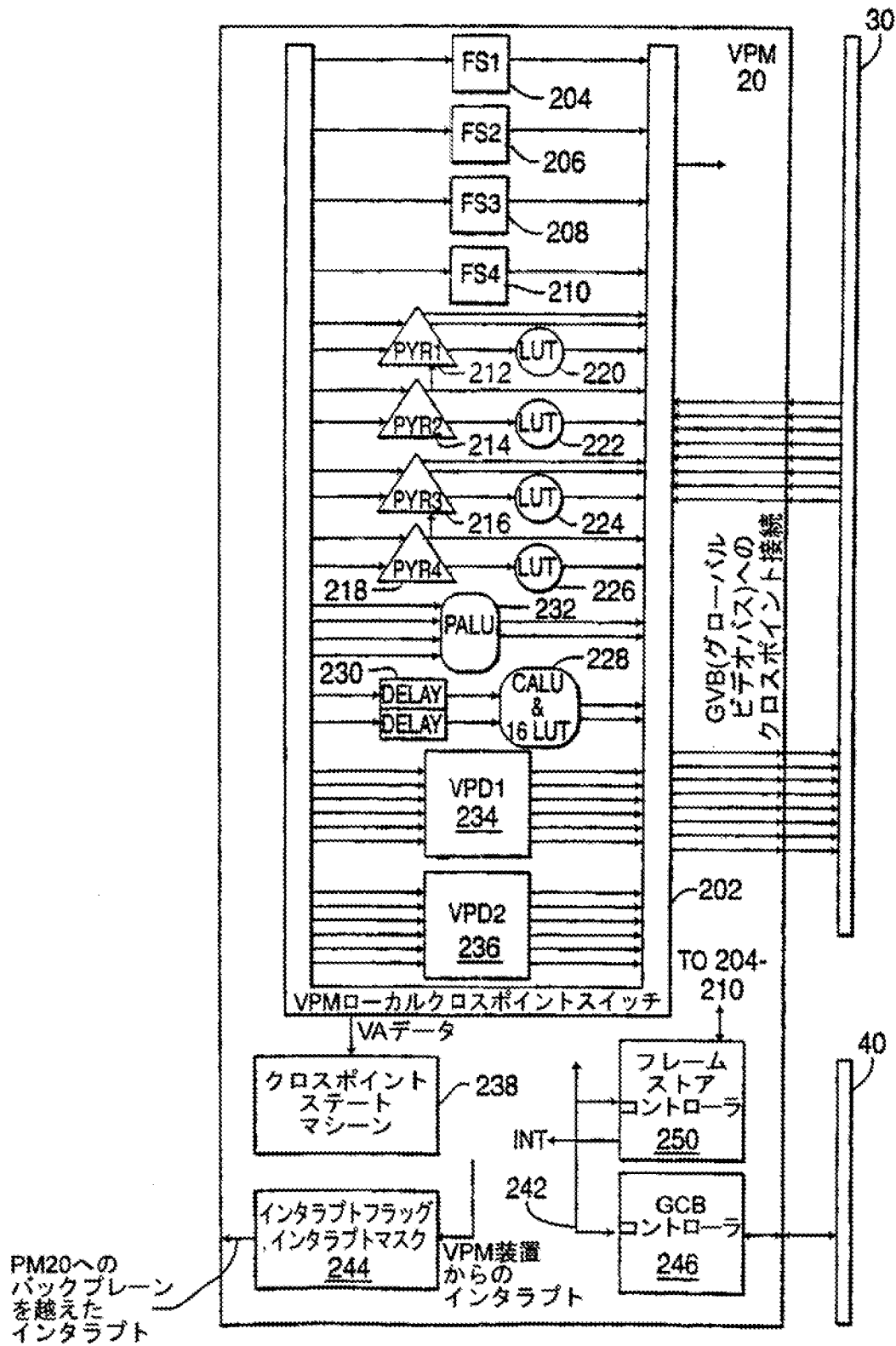
【図8】

図8は、5つのビデオインプット及びアウトプットを含んでいるワーパVPDの好ましい実施形態の図である。

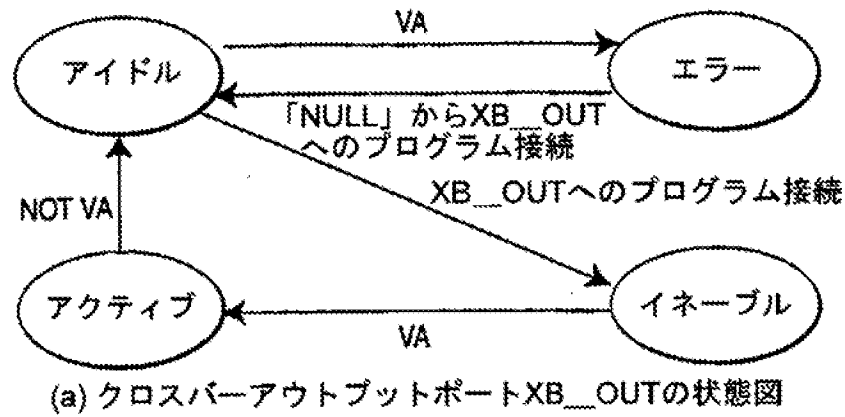
【図1】



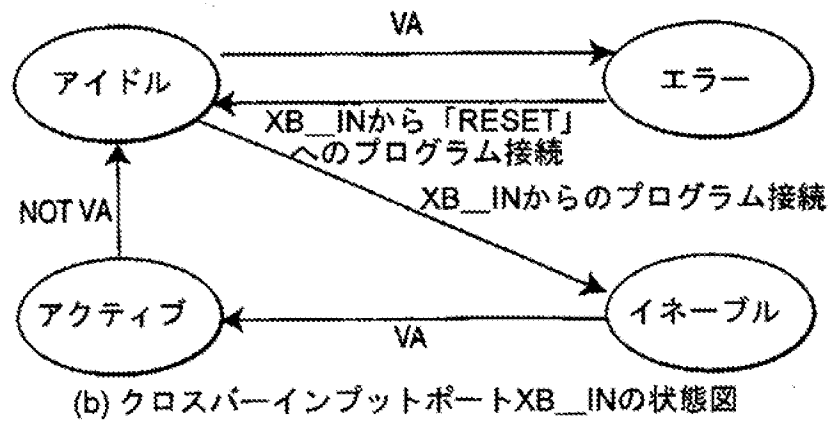
【図2】



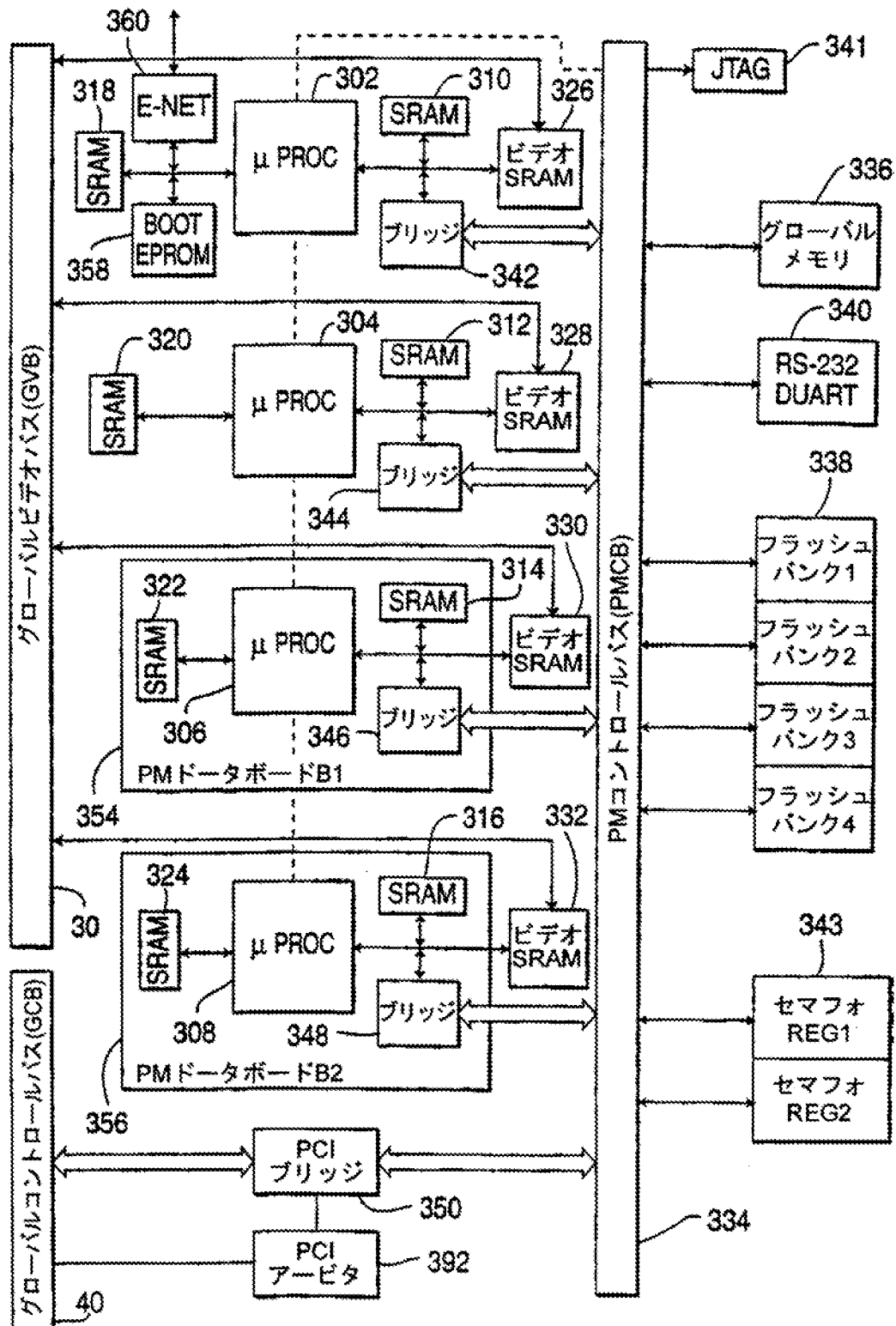
【図2 A】



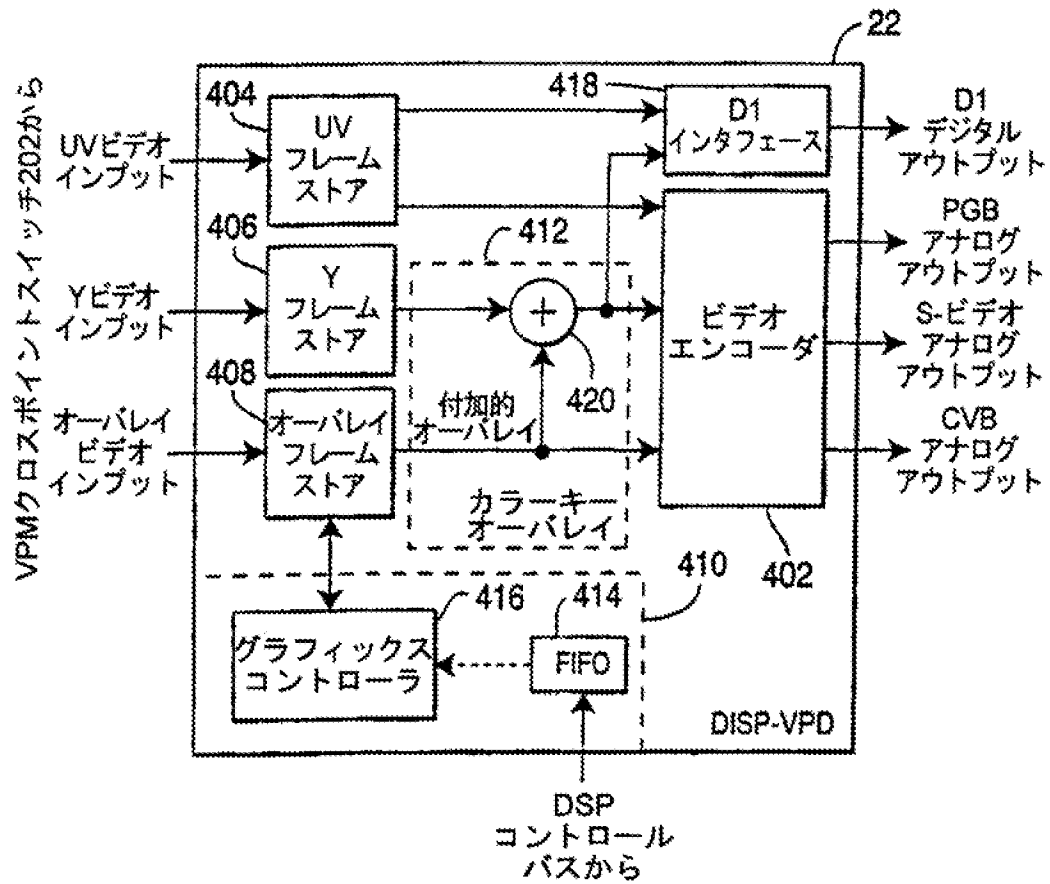
【図2 B】



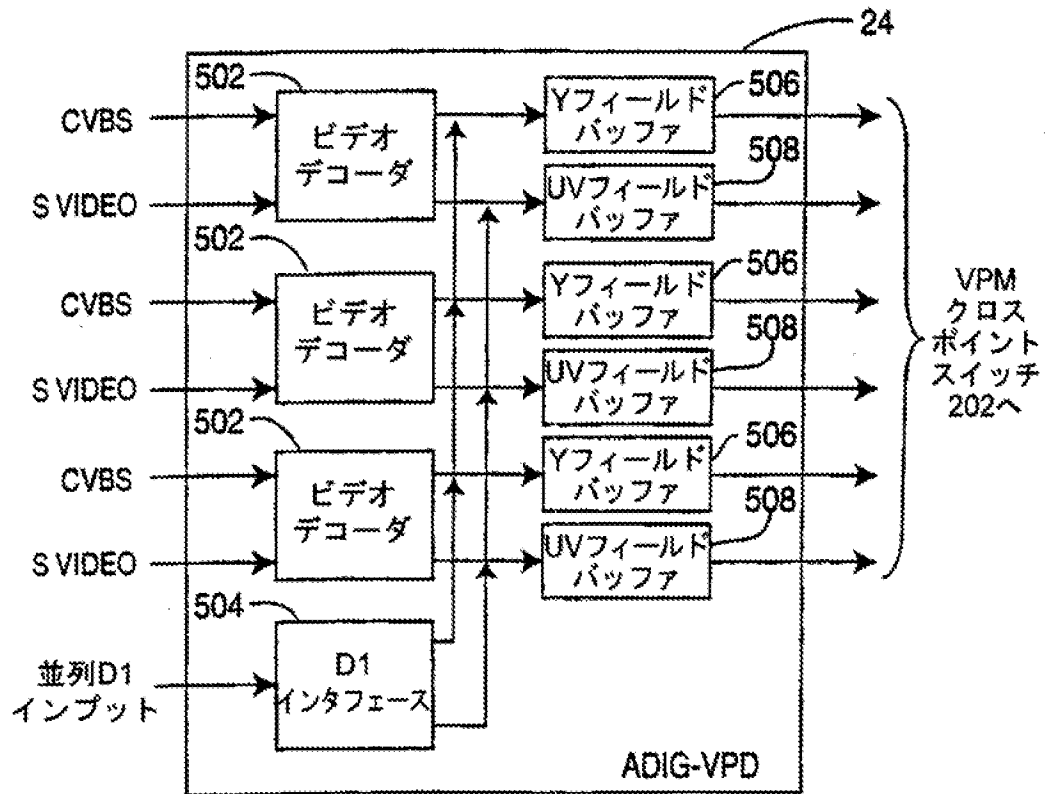
【図3】



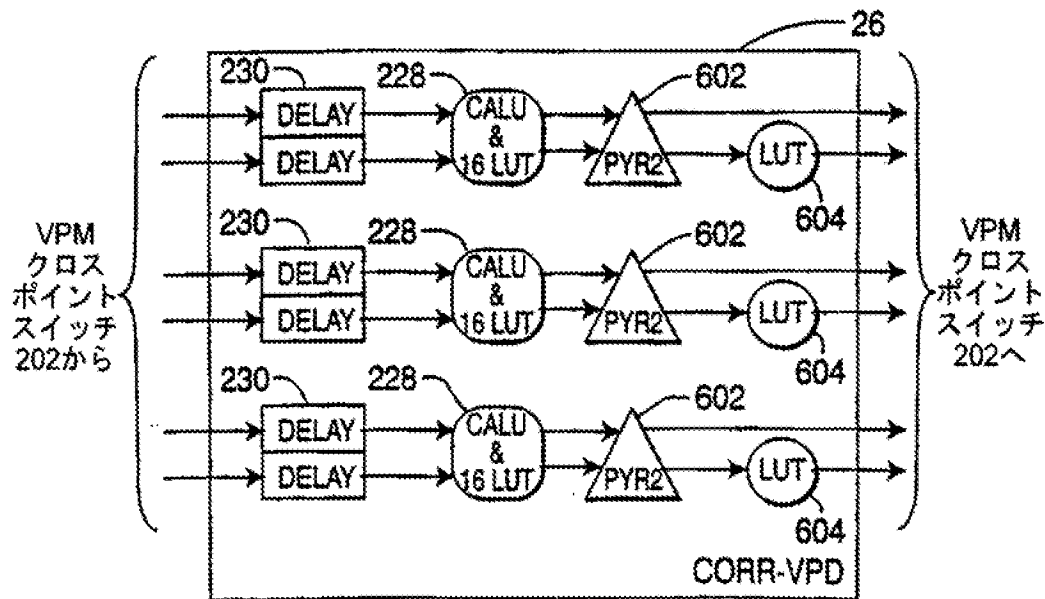
【図4】



【図5】

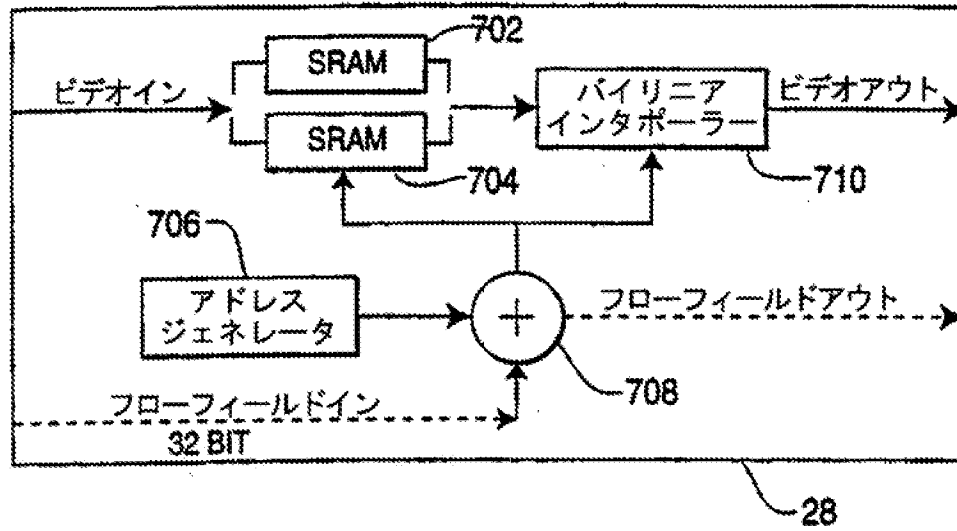


【図6】

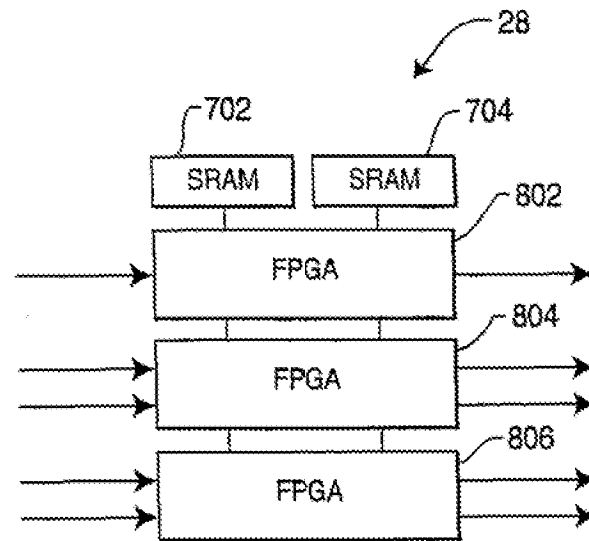




【図7】



【図8】

**FIG. 8**